

TARTU ÜLIKOOL
Arvutiteaduse instituut
Infotehnoloogia mitteinformaatikutele õppekava

Kersti Kivisoo

**Pythoni programmeerimise algõppe e-kursuse
„Programmeerimise alused”
läbiviimine kombineeritud õppena koolis**

Magistritöö (15 EAP)

Juhendajad:

Tauno Palts, MA

Merilin Säde, MA

Tartu 2018

Pythoni programmeerimise algõppe e-kursuse „Programmeerimise alused” läbiviimine kombineeritud õppena koolis

Lühikokkuvõte:

Programmeerimise õpetamine kogub aina rohkem populaarsust erinevates vanuseastmetes. Selles töös vaadeldakse, milliseid eestikeelseid programmeerimise õppematerjale saaksid õpetajad kasutada koolis programmeerimise kursuse läbiviimisel. Magistritöö eesmärgiks on koostada, toetudes Tartu Ülikooli programmeerimise e-õppematerjalidele, õppematerjalid Pythoni programmeerimise algõppe õpetamiseks kombineeritud õppena kolmandas kooliastmes ja gümnaasiumis. Õppematerjale katsetati ning kohendati vastavalt programmeerimise valikkursuse reaalsele läbiviimisele. Selleks viidi kaheteistkümne nädala jooksul üldhariduskoolis läbi programmeerimise algõppe kursus “Programmeerimise alused” kombineeritud õppena. Kursuse läbiviimisel kasutati Tartu Ülikooli arvutiteaduse instituudi poolt loodud õpilaste MOOC-i „Programmeerimise alused” õppematerjale ning Moodle’i õppekeskkonda. Toimunud tundide jaotuse ja lahendatud ülesannete järgi koostati programmeerimise aluste õpetamiseks ainekava ning tööplaani õpetajale kasutamiseks. Selle paremaks õnnestumiseks analüüsiti õppijate edasijõudmist kursusel ning õppijate antud hinnanguid kursusele, mille ülevaade on diagrammidena esitatud töös. Läbiviidud uuring näitas, et koolis saab programmeerimise õpetamisel MOOC-i materjale edukalt kombineerida kasutades ümberpööratud klassiruumi elemente, kuid silmas tuleb pidada mõningaid erisusi võrreldes vaba juurdepääsuga e-kursustega. Tehtud töö pakub ühe lahenduse, kuidas programmeerimise algõppe kursust koolis läbi viia.

Võtmesõnad: Kombineeritud õpe, ümberpööratud klassiruum, MOOC, programmeerimise algõpe

CERCS: S281 Arvuti õpiprogrammide kasutamise metoodika ja pedagoogika

Implementing Python Introductory Programming e-Course "Introduction to Programming" as a Blended Learning at School

Abstract:

Teaching programming is gaining more and more popularity among different age groups. This paper examines which teaching materials for teaching programming in the Estonian language teachers to conduct a course in school could use. The aim of the Master's thesis is to create the teaching materials for teaching programming in Python as a blended course on the third level of basic school and upper secondary school based on the materials of the Tartu University for programming. Study materials were tested and adjusted according to the realization of the course. The "Introduction to Programming" course was introduced during twelve weeks as a combined course in secondary and upper secondary school for this purpose. The course was conducted using the Moodle Learning Environment and study materials for the students' MOOC "Programming Basics" created by the University of Tartu Institute of Computer Science. Based on the division of tasks and the programming tasks, a syllabus and a work schedule for the teacher was created. In order to succeed better, the students' progress in the course and the given assessments were analyzed, the overview of which is presented in diagrams. The conducted study showed that teaching programming in school can successfully blend MOOC materials using flipped classroom elements, but there are some differences in terms of free e-courses. The work done offers one solution to how to teach programming course at school.

Keywords: Blended learning, Flipped classroom, MOOC, introductory programming

CERCS: S281 Computer-assisted education

Sisukord

1. Sissejuhatus	6
2. Programmeerimise algõpe ja kombineeritud õpe	8
2.1 Programmeerimise õpetamisest	8
2.1.1 Programmeerimiskeele Python õpetamine	10
2.1.2 Eestikeelsed Pythoni õppematerjalid	11
2.1.3 Programmeerimise õpetamise meetodid	13
2.2 Programmeerimise e-kursustest	14
2.3 Kombineeritud õpe ja meetodid	16
2.3.1 Kombineeritud õpe	17
2.3.2 Kombineeritud õppe kasutamine ja väljakutsed	18
2.3.3 Kombineeritud õpe ja MOOC	24
3.4.4 Kombineeritud õpe ja ümberpööratud klassiruumi mudel	25
3. Metoodika	27
3.1 Taust	27
3.2 Ainekava arendus	28
3.3 Kursuse uurimine	30
4. Tulemused ja arutelu	32
4.1 Ainekava ja kursuse materjalide koostamine	32
4.1.1 Ainekava	32
4.1.2 E-õpe	37
4.1.3 Tegevused auditoorses tunnis	38
4.2 Kursuse uurimine ja analüüs	41
4.2.1 Küsitluste analüüs	41
4.2.2 Ülesannete lahendamise analüüs	50
4.3 Arutelu ja edasised soovitused	54
5. Kokkuvõte	61
6. Kasutatud kirjandus	65
Lisad	73
I. Kursuse „Programmeerimise alused” läbiviimise plaan ja ajakava	73
II. Kursuse „Programmeerimise alused” ainekava	79
III. Kursuse „Programmeerimise alused” töökava	81
IV. Kursuse avaküsitluse küsimustik	90

V. Kursuse lõpuküsitluse küsimustik	92
VI. Programmeerimisülesannete tabel	95
VII. Lihtlitsents	98

1. Sissejuhatus

Eesti infoühiskonna arengukavas 2020 käsitletakse IKT (info- ja kommunikatsiooni-tehnoloogia) sektorit kui Eesti juhtivaimat majandusharu. Selleks, et Eesti saaks liikuda kõrgema lisandväärtusega töökohtade ning konkurentsivõime poole, oodatakse muuhulgas Eesti haridussüsteemilt IKT-valdkonna õppijate arvu suurendamist ja õppekvaliteedi parandamist (Eesti infoühiskonna arengukava 2020, s.a.). Lisaks tuuakse Eesti elukestva õppe strateegias 2020 välja, et uue põlvkonna digitaristu ja selle kasutamise meetodikad loovad võimaluse uue õpikäsitluse kiireks juurutamiseks ning õppekvaliteedi tõusuks (Eesti elukestva õppe ..., 2014). Rõhutatakse, et digitaalsete õppematerjalide kasutamine õppetöös aitab õppimist kõitvamaks muuta ning kogu elanikkonna parem tehnoloogia kasutamise oskus aitab kaasa majanduse kasvule. Pedaste jt (2015) sõnul on IKT-valdkonna arenguks vaja muuta praegu antavat infotehnoloogia haridust.

Koolidelt oodatakse ühiskonna ootustega kohanemist. Eesti õpetaja ametialane kohustus on suuta tegutseda muutivas haridussituatsioonis kasutades õpilase individuaalset arengut toetavaid õpetamismeetodeid (Õpetajate koolituse raamnõuded, 2015). Õppesisu valikul ja õppekavade kaasajastamisel on vaja arvestada pidevalt muutuvat haridusnõudlust. Selleks, et tekitada noortes huvi infotehnoloogia valdkonnaga seotud karjääri vastu, on oluline kvaliteetne infotehnoloogia alane õpe. Programmeerimisoskuse vajalikkusele viitab Eesti Valitsuse poolt koostatud Eesti Euroopa Liidu poliitika 2015–2019 raamdokument (Eesti Euroopa Liidu ..., 2015), kus tuuakse esile infotehnoloogiaalase hariduse edendamise olulisus, et lahendada majanduse ja ühiskonna probleeme. Muuhulgas väidetakse selles raamdokumendis, et juba varases eas on tarvis toetada ja soodustada programmeerimisoskuse arendamist ning samuti tähtsustatakse mitmekesise digitaalse õppevara kättesaadavust ja valikut.

Eesti riiklik õppekava ei näe ette kohustuslikke informaatikaõpetuse tunde, kuid samas kohustab kooli arendama õpilaste digitaalset pädevust (Põhikooli riiklik õppekava, 2011; Gümnaasiumi riiklik õppekava, 2011). Programmeerimise õpetamise laiendamiseks lisati 2013. aastal rakendunud Eesti gümnaasiumi riiklikku õppekavasse valikkursus „Rakenduste loomise ja programmeerimise alused”, kuid nagu Praxise uuringust selgus, õpetatakse seda kursust vaid 16% gümnaasiumiharidust pakkuvas koolis. Siiski on 2016. aastal avaldatud uuringu „Muutunud õpikäsitluse rakendamise ja selleks erinevate valikute pakkumise seire korraldamine, sh gümnaasiumis valikkursuste rakendamise osas” põhjal õppijad avaldanud

soovi õppida programmeerimise valikkursusi (Lamesoo & Ader, 2016). Praxise uuringu tulemuste põhjal vajavad õpetajad digioskuste õpetamisel (sh programmeerimise õpetamine) peamiselt kvaliteetsete digitaalsete õppematerjalide head kättesaadavust ja valikut (Leppik *et al.*, 2017). Seega õppijad tahaksid õppida programmeerimist, kuid õpetajad vajavad kvaliteetseid õppematerjale.

Tartu Ülikooli arvutiteaduse instituudi programmeerimise õpetamise töörühm on välja töötatud eestikeelsed programmeerimise algõppe materjalid programmeerimiskeele Python õpetamiseks e-õppe kursustena. Erinevaid e-õppe kursuste materjale on kasutatud kombineerituna auditoorse õppega nii programmeerimise õpetamisel kui teistes valdkondades (Hadjerrouit, 2008; Bruff *et al.*, 2013; Israel, 2015; Chen *et al.*, 2017). Püsib küsimus, kuidas ja millises osas saab väljatöötatud eestikeelseid programmeerimise õppimise materjale kasutada koolides programmeerimise valikkursuse läbiviimisel. Kombineeritud õppe meetod ei ole töö autori andmetel Eesti koolides programmeerimise õpetamisel laialdaselt levinud.

Selle töö eesmärk on koostada õppematerjalid ning kirjeldada nende kasutamise võimalust Pythoni programmeerimise algõppe kombineeritud õppena läbiviimisel kooli kolmandas kooliastmes või gümnaasiumiastmes valikkursusena kasutades Tartu Ülikooli e-õppe kursuse „Programmeerimise alused õpilastele” materjale.

Magistritöö koosneb kolmest osast. Töö esimeses osas antakse toetudes erinevatele uuringutele ülevaade programmeerimise õpetamisest Eesti koolides ja olemasolevatest eestikeelsetest e-õppematerjalidest, tutvustatakse programmeerimise õpetamise võimalusi läbi e-kursuste ning arutletakse kombineeritud õppe kasutamise võimaluste üle programmeerimise algõppe läbiviimisel. Töö teises osas antakse ülevaade tehtud uuringu metoodikast. Kolmandas osas antakse ülevaade läbiviidud programmeerimise algõppe kursuse ainekava koostamisest, õppematerjalide ja -meetodite kasutamisest, analüüsitakse kursuse õppijate hinnanguid kursusele ning arutletakse kursuse parendamise võimaluste üle.

Töö lisades on toodud koostatud kursuse „Programmeerimise alused“ ainekava ja töökava, ülevaade kursuse läbiviimise plaanist ja realiseerunud tegevustest, viited kursusel kasutatud programmeerimisülesannetele ning samuti kasutatud avaküsitluse ja lõpüküsitluse küsimused.

2. Programmeerimise algõpe ja kombineeritud õpe

Selles peatükis antakse toetudes erinevatele uuringutele ülevaade programmeerimise õpetamise olukorrast Eesti koolides, olemasolevatest eestikeelsetest programmeerimiskeele Python õppematerjalidest ja e-kursustest ning arutletakse, kuidas olemasolevaid e-õppe materjale kasutada programmeerimise kursuse läbiviimisel koolis. Samuti antakse ülevaade kombineeritud õppest, selle väljakutsetest ning kasutamise võimalustest programmeerimise õpetamisel.

2.1 Programmeerimise õpetamisest

Koolidelt oodatakse õpilaste ettevalmistamist ettevõtlikeks ja infoühiskonnas teadlikeks kodanikeks, millega kaasneb suutlikkus kasutada uuenevat digitehnoloogiat ning erinevaid õpistrateegiaid (Põhikooli riiklik õppekava, 2011; Gümnaasiumi riiklik õppekava, 2011). Järjest enam rõhutatakse programmeerimisalaste teadmiste olulisust tänapäeva ühiskonnas edukalt hakkama saamiseks (Ferrar, 2013; Eesti Infoühiskonna arengukava 2020, s.a.). Programmeerimise õpetamisel rakendatakse nii traditsioonilisi kui konstruktivistlikke õpistrateegiaid. Traditsioonilises õpikäsitluses on õppimine orienteeritud aine sisule, ülesanded on valdavalt teoreetilised ning rõhuasetus ei ole uute teadmiste seostamisel varem õpituga (Smit *et al.*, 2014). Nüüdisaegne õpikäsitus tugineb konstruktivistlikule lähenemisele ning selle korral omandab õppija teadmisi läbi enda varasema kogemuse ning õppimine on tihedalt seotud sotsiaalse suhtluse ja koostööga (Heidmets, 2017; de Kock *et al.*, 2004). Arusaam õppimisest ja õpetamisest mõjutab seda, kuidas õppija õpib. Cheang (2009) väidab, et õppijakeskne õpikäsitus tõstab õppija motivatsiooni õppimisele. Kõrgem motivatsioon mõjutab omakorda õppimise edukust. Õpitav peab olema õppijale huvitav ning võiks olla seotud kaasaegsete tehnoloogiliste võimaluste kasutamisega, milleks on tarvis tagada õpetajale õpet toetavad materjalid ja e-keskkonnad.

Põhikooli ja gümnaasiumi riiklikes õppekavades ei ole kohustuslikku õppeainet infotehnoloogia alaste teadmiste ja oskuste õpetamiseks (Põhikooli riiklik õppekava, 2011; Gümnaasiumi riiklik õppekava, 2011). Programmeerimise pädevuse arendamine on kohustuslik vaid neile koolidele, kes õpetavad programmeerimist eraldi aina (Mets jt, 2016). Põhikooli riiklikus õppekava valikaines „Informaatika” on lähtutud igapäevase arvuti- ja internetikasutaja vajadustest (Põhikooli riiklik õppekava, 2011). Gümnaasiumi riiklikus õppekavas ei ole eraldi infotehnoloogia ainevaldkonda, loodusainete ainevaldkonnas on

valikkursus „Rakenduste loomise ja programmeerimise alused” ning selle kursuse jaoks on koostatud õppematerjalid (Gümnaasiumi riiklik õppekava, s.a.). Praxisse uuring (Leppik *et al.*, 2017) näitas, et koolides puudub digioskuste õpetamist toetav ühtne süsteem ning koolid otsustavad ise, kuidas nad infotehnoloogiat õpetavad. Vaid 1% üldhariduskoolidest on oma ainekavades viidanud programmeerimisega seotud tegevustele, infotehnoloogia alaseid huviringe on pakkunud vähemalt pooled üldhariduskoolid, programmeerimise huviringe on huviringidest 39% (Leppik *et al.*, 2017). Samuti näitab Siim Puniste (2014) koostatud bakalaureusetöö „Eesti gümnaasiumides õpetatavad programmeerimiskursused” tulemus, et Eesti üldhariduskoolides õpetatavad kursused erinevad üksteisest sisu poolest ning riikliku õppekava valikkursuse materjale kasutatakse vähe. Näiteks toob Puniste välja, et uuritud kursustel oli kasutusel 14 erinevat programmeerimiskeelt, teemade käsitluste sügavused ning kasutatavad õppematerjalid erinesid kooliti. Valikkursuse head õpetamist iseloomustab õpilaste arvates see, kui on kasutatud mitmekesiseid õppemeetodeid ning häid õppe- ja juhendmaterjale (Lamesoo & Ader, 2016). Siiski ei jagu koolides Pedaste jt (2015) väitel programmeerimise algteadmiste õpetamiseks hiljemalt gümnaasiumiastmes pädevaid õpetajaid ning paljud praegused infotehnoloogia ainete õpetajad ei oma infotehnoloogiaalast kõrgharidust.

Kori jt (2015a) väidavad, et koolide ebaühtlase infotehnoloogia alase hariduse tõttu puutub enamus Eesti infotehnoloogia eriala üliõpilasi, ligikaudu 56%, programmeerimisega esimest korda kokku alles ülikoolis, kus paljude üliõpilaste jaoks rasked programmeerimise algkursused on seotud üliõpilaste ülikoolist väljalangevusega. Dixon (2013) väidab, et kuigi ülikooli õppima tulijad on kokku puutunud kaasaegse tehnoloogiaga, on ekslik oletada, et neil on olemas kõik õpingute lõpetamiseks vajalikud infotehnoloogia alased oskused. Ülikooli jaoks on probleemiks suur, Eestis 32,2%, infotehnoloogia eriala õppijate väljalangevus esimesel aastal (Kori *et al.*, 2015b). Seetõttu on ülikoolid huvitatud, et informaatika eriala sisseastujad omaksid juba eelnevaid teadmisi sellest, mida nad õppima hakkavad. Sisseastujalt ei nõuta eelnevat programmeerimise oskust, kuid väljalangejate arvu vähendamaks sisseastujate teadmine vastava eriala sobivusest (Kori *et al.*, 2015b). Pedaste jt (2015) annavad koolidele soovitusi varakult alustada kvaliteetse infotehnoloogia õppega, muuhulgas anda ka õpilastele võimalus õppida programmeerimist.

2.1.1 Programmeerimiskeele Python õpetamine

Maailmas on palju erinevaid programmeerimiskeeli. Programmeerimiskursuse sisulise osa üheks tähtsamaks küsimuseks on õpetatava programmeerimiskeele valik (Leping *et al.*, 2009). On oluline, et esimene õpitav programmeerimiskeel jätaaks õppijale positiivse mulje ning motiveeriks programmeerimist edasi õppima (Ateeq *et al.*, 2014). Samuti väidavad uuringud, et esimene programmeerimiskeel mõjutab õppija edaspidist programmeerimise stiili ja tehnikat (Leping *et al.*, 2009; Ateeq *et al.*, 2014).

Algajatele programmeerimise õpetamisel on tänapäeval üheks levinumaks programmeerimiskeeleks Python. Vabavaraline programmeerimiskeel Python, mis töötab erinevates operatsioonisüsteemides, on kerge loetavusega, eriti lihtsate programmide puhul, kuna sarnaneb inglise keelega. Seda näitab samuti joonisel 1 toodud programmeerimiskeeltes Python ja Java kirjutatud programm, mis väljastab ekraanile sõnapaari „Hello, World”.

Python: <pre>print("Hello, World")</pre>	Java: <pre>public class HelloWorld { public static void main(String[] args) { System.out.println("Hello, World"); } }</pre>
---	--

Joonis 1. Programmi „Hello, World” võrdlus programmeerimiskeeltes Python ja Java.

Milbrandt pidas programmeerimiskeele õppimise juures lisaks programmeerimiskeele süntaksi lihtsusele muuhulgas oluliseks ka kiire tagasiside andmist õppijale (viidatud Stephenson, 2000). Tagasiside võib seisneda nii juhendaja arvamuses, programmi edukas või mitteedukas kompileerimises. Aivar Annamaa on arendanud programmeerimiskeele Python jaoks vabavaralise programmeerimiskeskonna Thonny. See on vahend, mille abil visualiseeritakse õppija jaoks programmi töökäik sammhaaval ning on tagasiside andmisel õppijale suureks abiks. Thonny eeliseks Pythoni programmeerimiskeskonna IDLE (ingl *Integrated Development Environment*) ees on see, et käsurida asub koodiredaktoriga samas vaates ning on seetõttu lihtsamini jälgitav. Thonny on mõeldud eelkõige programmeerimise õppimiseks ja õpetamiseks, kuna funktsionaalsus, mis näitab kasutajale programmi tööd sammhaaval (ingl *debug*), aitab õppijal mõista, kuidas programm täpselt töötab (Annamaa, 2015). Samuti on üheks Thonny funktsionaalsuseks kasutaja poolt programmiteksti kirjutamisel tehtud toimingute, näiteks teksti sisestamine, kasutamine ja mujalt kopeeritud teksti kleepimine, logimine (Aramaa, 2014). Thonny logifailide informatsioon annab teavet

õppija tegevuste kohta ülesannete lahendamisel, mis aitab õpetajal näiteks leida neid õppijaid, kes vajavad ülesannete lahendamisel abi.

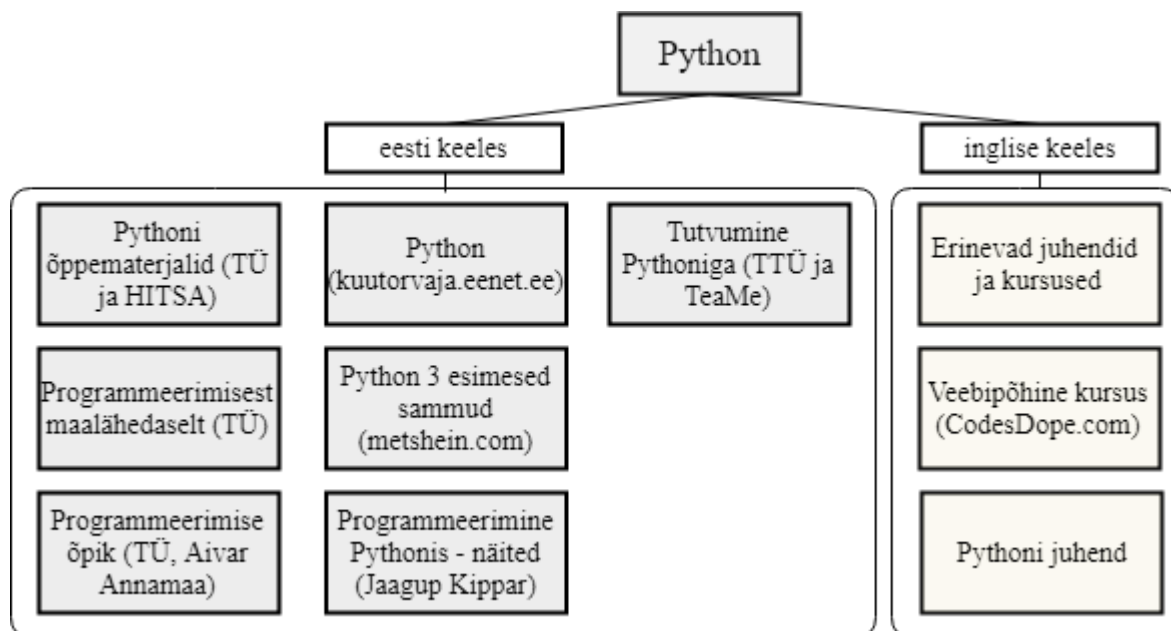
Kuna Python on populaarne (TIOBE Index, 2018), arenev keel ja laialdaselt kasutatav ka ettevõtetes (Python, 2018), siis on selle õppimiseks veebis palju erinevaid võimalusi ja materjale, millele toetuda. Siiski on õpetaja ja õppija jaoks koolis on oluline terviklike ja süsteemselt koostatud eestikeelsete õppematerjalide olemasolu. Tartu Ülikoolis on esimese õpetatav programmeerimiskeel Python (Leping *et al.*, 2009). Samuti väidavad uuringud, et Python sobib hästi esimeseks programmeerimiskeeleks õpilastele (Mannila & de Raadt, 2006).

2.1.2 Eestikeelsed Pythoni õppematerjalid

Codecademy veebipõhisel platvormil leiab õppematerjale, kus saab omal käel õppida üheksat erinevat programmeerimiskeelt (ProgeTiiger, s.a.). Siiski ei ole sealsed programmeerimiskeele Python tehnilisi oskusi õpetavad terviklikud kursused eestikeelsed.

Aastal 2012 alustas Tiigrihüppe Sihtasutus koolidele suunatud programmiga ProgeTiiger, mille raames saavad erinevas vanuses õppijad teha tutvust programmeerimisega. ProgeTiiger pakub eelkõige eestikeelsete materjalide kogumikku programmeerimise jaoks ning on mõeldud õpetajatele õppetöös kasutatavate töövahendite leidmiseks. Otsida saab programmeerimiskeeli PHP, JavaScript, Python, Java, Ruby, jQuery, HTML/CSS, Logo/UBCLogo kasutatavate õppematerjalide hulgast. Kogumikus (ProgeTiiger, s.a.) on viited kümnele programmeerimiskeelt Python õpetamise materjalile (materjalide ülevaade on kujutatud joonisel 2), neist seitse on eestikeelsed materjalid Pythoni keele õpetamiseks:

1. Python koolis (s.a.), Tartu Ülikooli õppematerjalid huviringides programmeerimise õpetamiseks kasutades Pythoni moodulit Pygame. Õppematerjal koosneb 32 ringitunni materjalist, mille rõhk on eelkõige probleemilahendusel ja iseseisval töötlahenduste otsimisel.
2. Aivar Annamaa programmeerimise õpik (Annamaa, s.a.), mis on koostatud silmas pidades eelkõige informaatika bakalaureuse õppekava programmeerimise algkursust. Eelteadmisi programmeerimisest õpik ei eelda ning sobib seetõttu ka iseõppijale.

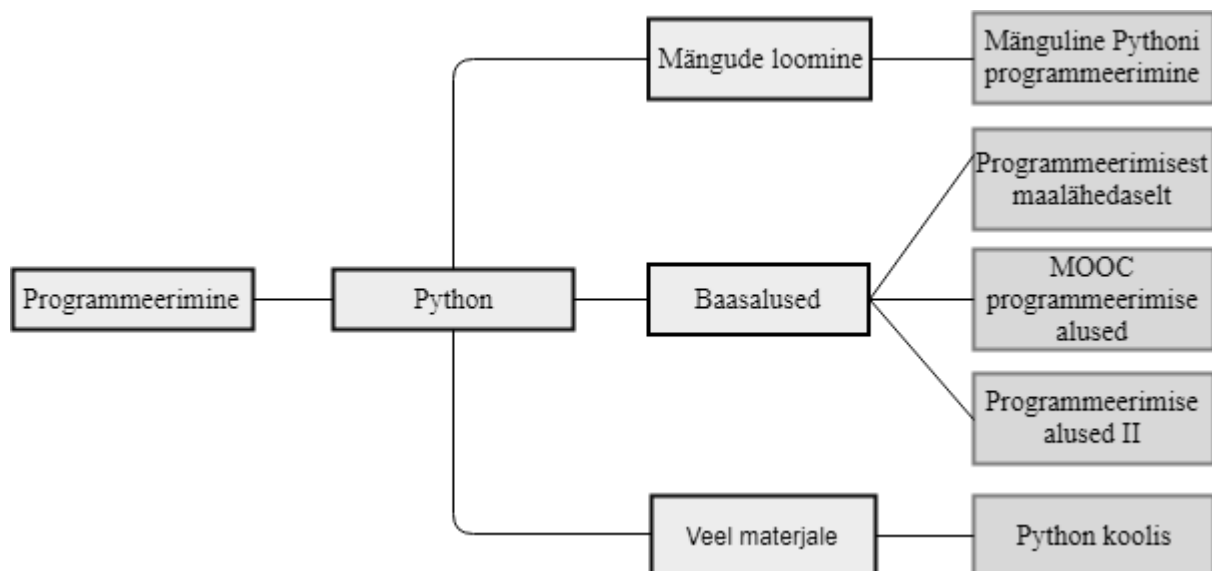


Joonis 2. ProgeTiiger Pythoni õpetamiseks esitatud materjalid (ProgeTiiger, s.a.).

3. Programmeerimisest maalähedaselt (s.a.), Tartu Ülikooli vastava tasuta täieliku veebipõhise kursuse materjalid. Kursuse eesmärgiks on tutvustada programmeerimist ja sellega seonduvat neile, kel programmeerimisega varasemat kokkupuudet pole olnud. Kursus on täielikult e-kursus, mille ametlik osa on Moodle'i keskkonnas.
4. Kuutõrvaja Pythoni õppematerjalid (s.a.). Kuutõrvaja EENet eestikeelsed materjalid programmeerimise kohta on mõeldud arvuti tavakasutajale.
5. Metshein Pythoni algkursuse materjalid, Python 3 esimesed sammud (s.a.). Kursuse eesmärk on anda algteadmised programmeerimisest Python 3 baasil. Eesmärgiks pole niivõrd just keele õppimine, vaid mõtteviisis, kuidas asjad tööle hakkavad.
6. Programmeerimine Pythonis, erinevad näited, Jaagup Kippari materjalid (s.a.). Tegemist on erinevaid Pythoni koodinäiteid sisaldava veebilehega.
7. Tutvumine Pythoniga (s.a.). Materjal on Tallinna Tehnikaülikooli töörühma poolt loodud gümnaasiumile mõeldud kursuse „Rakenduste loomise ja programmeerimise alused” osa, mis asub HITSA (Hariduse Infotehnoloogia Sihtasutus) Moodle-i õpikeskkonnas. Õppematerjali Pythoni osa ei ole sobiv programmeerimise õppimiseks eraldi materjalina, kuna programmeerimise mõisted õpitakse eelnevalt kasutades Scratchi osa.

Lisaks ProgeTiigri kogumikule on Tartu Ülikooli poolt loodud ja läbiviidavate erinevate programmeerimisalaste kursuste materjalid kogutud kokku infotehnoloogia alaste kursuste

materjalide veebilehele (Tartu Ülikooli IT-kursuste ..., s.a.). Materjalide ülevaade on toodud joonisel 3.



Joonis 3. Tartu Ülikooli IT-kursuste lehel olevad programmeerimise kursuste materjalid (Tartu Ülikooli IT-kursuste ..., s.a.).

Baasaluste õppimiseks on täiskasvanutele mõeldud kursused: „Programmeerimine maalähedaselt” (samale materjalile viitab ka ProgeTiigri materjal), „Programmeerimise alused” ja „Programmeerimise alused II”. Lastele ja noortele on mõeldud kursus „Mänguline Pythoni programmeerimine”, gümnaasiumiastme noortele on mõeldud lisaks kursus „Python koolis” (samale materjalile viitab ka ProgeTiigri materjal). Erinevaid õppematerjale on veel mitmeid, töös on toodud vaid mõned näited. Seega eestikeelseid materjale programmeerimiskeele Python õppimiseks ja õpetamiseks on olemas, kuid küsimus on selles, kuidas neid materjale saab kasutada programmeerimise valikkursuse läbiviimisel.

2.1.3 Programmeerimise õpetamise meetodid

Õpilastel võiks olla võimalus õppida infotehnoloogia alaseid teadmisi ja oskusi oma koolis, huviringis või veebis. Praxise (Leppik *et al.*, 2017) uuringust selgus, et 63% õpetajatest ei ole end viimase kahe aasta jooksul täiendanud tehnoloogiahariduses (sh programmeerimine), kuid samas on 43% õpetajatest soov enesetäiendamiseks. Seega huvi programmeerimise õppe vastu on olemas ka õpetajate hulgas, mis omakorda aitab kaasa pädevate õpetajate arvu suurenemisele koolides, kui motiveerida õpetajaid enesetäiendamisele.

Puniste (2014) toob välja, et õpetatava programmeerimiskeele valikul on oluline õpetaja enda kogemus antud keelega. Programmeerimise õpetamise väljakutseks on erinevate uuenduslike õpetamismeetodite rakendamine, et tõsta õppijate edukust ning suurendada õpimotivatsiooni (Mohorovičić & Stričić, 2011). Nurk (2015) soovib oma tehtud tööle tuginedes kasutada gümnaasiumiastmes paarisprogrammeerimise (ingl *pair programming*) meetodit. Paarisprogrammeerimine on programmeerimise viis, kui kaks programmeerijat lahendavad sama ülesannet üheskoos ühe arvuti taga ning arutelu võib aidata teemat paremini mõista, kirjutatav kood võib olla ühtlasema disainiga ja sisaldada vähem vigu (Mohorovičić & Strčić, 2011).

Paljud õpetajad kasutavad programmeerimise õpetamisel segu enda loodud ja veebis kättesaadavatest materjalidest. Uuringud väidavad, et koolid vajavad kõrgkoolide abi programmeerimise alaste õppematerjalide koostamisel (Pedaste *et al.*, 2015; Puniste, 2014). Tartu Ülikool on viimastel aastatel arendanud programmeerimise õppimise veebikursuseid, mille materjale saab kasutada nii õppetundides kui ka iseseisval õppimisel. Loodud materjale saab õpetaja soovi korral kasutada programmeerimise tundides, kuid kuna vastavad kursused on Moodle'i toega, millele ei ole õpetajatel vaba juurdepääsu, siis ei ole sealsed materjalid õpetajatele kättesaadavad. Seega loodud materjalid on piirangutega ega ole kasutatavad sel ajal ega nii nagu õpetaja parajasti soovib.

2.2 Programmeerimise e-kursustest

MOOC (ingl *Massive Open Online Course*) tähendab avatud registreerimisega täielikult veebipõhist kaugõppe kursust (Jansen & Schuwer, 2015). Üldjuhul on MOOC tasuta, kuid leidub kursuseid, kus peab lõpetamise sertifikaadi saamiseks maksma (Shah, 2015). Samuti on kursustel tavaliselt ajapiirang, millal kursus toimub. MOOC-ide kaudu on võimalik end harida igal inimesel, kellel on juurdepääs internetiühendusele. Kuigi MOOC-id on mõeldud piiramatule hulgale inimestele ilma eelnevalt nõutavate teadmiste ja oskusteta, eeldab MOOC-il osalemine, et õppija on ise võimeline oma õppimist ja aega reguleerima (Littlejohn *et al.*, 2016). Lisaks on MOOC-ide kaudu seetõttu võimalik rakendada elukestva õppe strateegiat.

Tartu Ülikooli arvutiteaduse instituudi programmeerimise töörühm on viinud alates aastast 2014 läbi programmeerimiskeelt Python õpetava MOOC-i „Programmeerimine maalähedaselt”, mis oli ka ülikooli esimene eestikeelne MOOC (Lepp *et al.*, 2017). Kursus

on suunatud täiskasvanud inimesele, kellel puudub eelnev kokkupuude programmeerimisega ning käsitleb lihtsamaid programmeerimiskeele Python konstruktsioone ning avardab õppija silmaringi. Kursus on Eestis seni kõige suurema osalejate ja lõpetajate arvuga MOOC, mis valiti ka 2016. aasta parimaks e-kursuseks (Lepp *et al.*, 2017). Kursuse lõpetajate osakaal on ligikaudu 65% registreerunutest, mis on MOOC-ide puhul ebatavaliselt kõrge (Sarap, 2017). Kursuse e-õppe platvorm on Moodle, kus on võimalik kasutada ülesannete automaatkontrolliks VPL (ingl *the Virtual Programming Lab for Moodle*) liidest (Lepp *et al.*, 2017). Õppematerjalid on kättesaadavad Tartu Ülikooli arvutiteaduse instituudi õppematerjalide keskkonnas ehk Courses-keskkonnas ning neile on ligipääs kõigil huvilistel ka pärast kursuse lõppu.

Esimese kursuse suure populaarsuse tõttu loodi kursusele programmeerimisega põhjalikumaks tutvumiseks 2016. aastal kursus „Programmeerimise alused” (s.a.), mahuga 3 EAP ja kestusega 8 nädalat ning 2017. aastal lisaks veel eraldi samasisuline kursus gümnaasiumiõpilastele. Loodud uute kursuste ülesehitused olid esimese kursuse ülesehitusega sarnased, lisaks kursuse „Programmeerimisest maalähedaselt” (s.a.) teemade sügavamale käsitlemisele tutvustati ka järjendeid, kasutajaliideseid ja graafikat. Aasta 2017 kevadel alustas MOOC-ina ka kursuse järg „Programmeerimise alused II”, millel osalemine eeldab kursuse „Programmeerimise alused” teadmiste omamist ning kursusel süvendatakse algteadmisi programmide koostamisest, testimisest ja silumisest (Programmeerimise alused II, s.a.). Vastavad kursused on osutunud väga edukateks, mida näitab osalejate suur huvi kursuse vastu ning MOOC-ide tavalisest lõpetanute osakaalust kõrgem määr (Sarap, 2017).

Pedaste jt (2015) soovivad koolides tunnustada e-kursuseid kui ühte õppevormi. MOOC-i võib vaadelda kui iseseisvat e-kursust, kus kursusel osalemine ja selle läbimine on seotud õppija isikliku huviga või kui kursust, mis on osa haridusest ning õppija võib soovida hiljem kursuse läbimist arvestada osana õpingutest. MOOC-ide kaasamine haridusse on uudne kogu haridussüsteemis, kuid sellega kaasnevad mitmed väljakutsed, kuna enamasti pole teada, kas registreeritud osaleja lahendas ülesanded ise või kellegi teise abiga (Sarap, 2017). Arvestustöö läbiviimine kontrollitud oludes on küll võimalik, kuid raskendatud, kui kursuse osalejate arvud on suured. Siiani puudub ühtne kokkulepe MOOC-ide arvestamise kohta ja sellest tulenevalt on kooli enda otsustada, kuidas ja milliseid MOOC-e nad aktsepteerivad. Tartu Ülikool pakub võimalust üliõpilase esitatud taotluse alusel varasemaid õpinguid ja

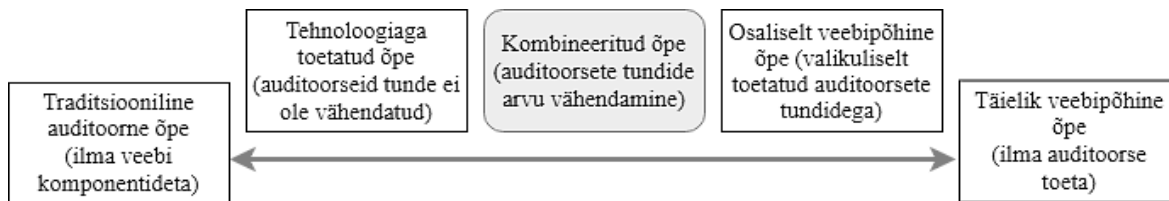
töökogemust arvestada (VÕTA) õppekava täitmisel (Tartu Ülikooli VÕTA ..., 2017). Gümnaasiumides sellist ühtset korda riiklikult sätestatud ei ole.

Projekti „Muutunud õpikäsituse rakendamise ja selleks erinevate valikute pakkumise seire korraldamine, sh gümnaasiumis valikkursuste rakendamise osas” raportis (Lamesoo & Ader, 2016) tuuakse välja, et üle poolte Eesti koolidest on arvestanud õppija huvihariduses omandatud pädevusi õppetöö osana, peamiselt valikkursusena. Igal koolil on võimalik läbi mõelda süsteem, mille alusel huviharidust valikainena arvestada, kui kool ise vastavateemalist valikkursust ei paku. Projektis selgus, et koolijuhid on valmis arvestama valikkursusena internetis läbiviidavaid kursuseid (Lamesoo & Ader, 2016). Raporti koostajad teevad Haridus- ja Teadusministeeriumile ettepaneku luua gümnaasiumide jaoks ühtne veebipõhiste valikkursuste süsteem, kus e-õppes saaks osaleda palju õpilasi olenemata nende elukohast. Programmeerimise MOOC-idel osalejate suured arvud näitavad, et huvi programmeerimise õppimise vastu on suur (Programmeerimise alused, s.a.).

Lisavõimaluseks e-kursuste kasutamisel on MOOC-ide kombineerimine tavakursustega, mida ülikoolis on juba tehtud, muuhulgas ka kursusel „Programmeerimise alused II” (Sarap, 2017). Lisaks täielikule MOOC-ile toimus 2017. aasta kevadel esimest korda kombineeritud õppevormina „Programmeerimise alused” kursus, kus üliõpilastel oli ligipääs samanimelise MOOC-i kursuse materjalidele veebilehel ja Moodle'i keskkonnas ning lisaks toimusid loengud ja praktikumid ning kursus toimus ümberpööratud klassiruumi mudeli järgi (Sarap, 2017).

2.3 Kombineeritud õpe ja meetodid

Interneti levik on aidanud kaasa distantsõppe (kaugõppe) populaarsuse kasvule. ning toonud kasutusele veebipõhise õppega seotud mõiste e-õpe, mida vastandatakse traditsioonilisele auditoorsele õppele. Õppe korraldamiseks ja läbiviimiseks on peale traditsioonilise auditoorse õppe ja täieliku veebipõhise õppe mitmeid erinevaid võimalusi, mis on toodud joonisel 4 (Graham *et al.*, 2013).



Joonis 4. Õppe läbiviimiste viisid (Graham *et al.*, 2013).

Üheks alternatiiviks on kombineeritud õppe kasutamine, mille alla sageli ekslikult arvatakse ka tehnoloogiaga toetatud õpe ja osaliselt veebipõhine õpe (Graham *et al.*, 2013).

2.3.1 Kombineeritud õpe

Kombineeritud õppe (ingl *blended learning*) mõiste võeti esmakordselt kasutusele Ameerika Ühendriikides ning selle all mõeldi traditsioonilise õppe ja kaasaegsel tehnoloogial põhineva õppe kombineerimist kasutades erinevaid pedagoogilisi meetodeid (Bonk & Graham, 2006). Kombineeritud õppe mõistel ei ole pedagoogilistes käsitlustes ühtset tähendust ja arusaama. Graham, Allen ja Ure dokumenteerisid kolm kõige enam kasutatavat kombineeritud õppe arusaama (Bonk & Graham, 2006):

- õppemeetodite (või sisu edastamise viiside) kombineerimine (ingl *combining instructional modalities (or delivery media)*);
- juhendamismeetodite kombineerimine (ingl *combining instructional methods*);
- veebipõhise ja auditoorse õppe kombineerimine (ingl *combining online and face-to-face instruction*).

Neist arusaamadest lähtuvalt defineeris Graham kombineeritud õppe kui auditoorse juhendamise ja arvuti abil vahendatud juhendamise kombinatsiooni (Bonk & Graham, 2006). Osguthorpe ja Graham jaotavad kombineeritud õppe kolmeks erinevaks mudeliks (Bonk & Graham, 2006):

- auditoorse õppe ja e-õppe õpitegevuste kombineerimine,
- auditoorset õpet ja e-õpet saavate õppijate kombineerimine,
- auditoorsete õpetajate ja e-õppe keskkonna õpetajate kombineerimine.

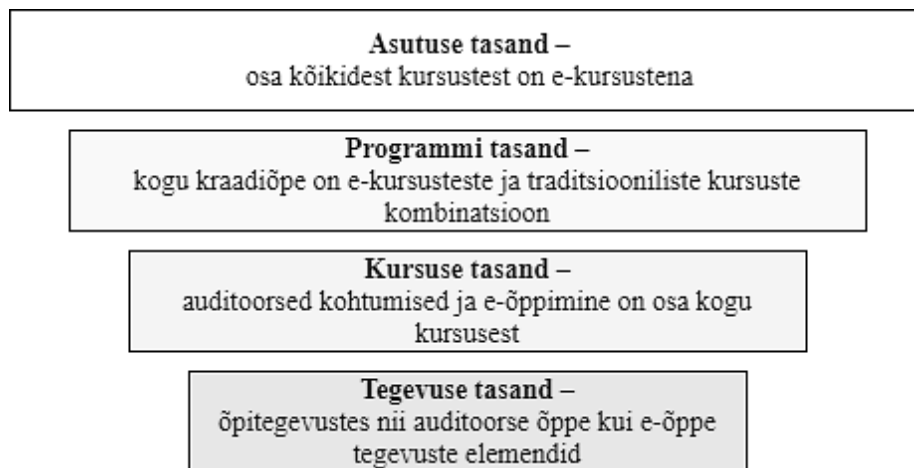
Enamasti mõistetakse kombineeritud õppe all õppimis- ja õpetamisviisi, mis ühendab eri meetodeid ja õpetamisvahendeid (traditsiooniline auditoorne õpe ühendatud innovaatiliste tehnoloogiliste vahenditega nagu näiteks veebipõhised vahendid, esitlus- ja koostöötarkvara, jms) (Kaur, 2012). Kaur (2012) toob välja kombineeritud õppe komponendid, milleks on õpikeskkond, juhendamine ja meedia ning nende omavahelise seotuse.

Õpikeskkond võib olla sünkroonne või asünkroonne, mille mõlema iseloomulikke edukaid komponente saab kasutada õpikeskkonna optimaalsuse saavutamiseks (Kaur, 2012). Meedia viitab igale vahendile, mis edastab ja toetab õppesisu. Uuringutes on leitud, et õppijad eelistavad õpetajale küsimusi esitada otse klassiruumis, mitte e-kirja kaudu (Güzer & Caner, 2013), mis näitab auditoorse õppe osa tähtsust. Seega on õpetaja juhendamine õppimise toetamisel esmatähtis.

Auditoorse ja veebipõhise aja suhe erineb kursuste lõikes, parimat suhet alles otsitakse. Graham, Woodfield ja Harrison (2013) väidavad, et on neljandikul kombineeritud õppe kursustel on asendatud 25% või rohkem auditoorseid tundide e-õppega. Israel (2015) väidab, et erinevate uuringute põhjal on kombineeritud õppe kasutamise korral veebipõhine osa 30% kuni 79% õpitavast. Mistahes kombinatsiooni puhul on oluline keskenduda oodatavatele õpitulemustele. On uuritud, kas traditsioonilise õppe või veebipõhise õppe puhul on õpilaste õpitulemused paremad (Kaur, 2012). Kombineeritud õpe annab paindlikkuse nii õppijatele kui õpetajatele ning võimalik on kasutada mõlema õppe häid külgi. Virtuaalse ja füüsilise keskkonna lõimimine võimaldab nii õppijatel kui õpilastel olla õppijad (Kaur, 2012). Väidetakse, et kombineeritud õpe on muutumas uueks traditsiooniliseks mudeliks (Graham *et al.*, 2013).

2.3.2 Kombineeritud õppe kasutamine ja väljakutsed

Kombineeritud õppimine võib toimuda Grahami, Woodfieldi ja Harrisoni (2013) väitel erinevatel tasanditel (joonis 5): tegevuse tasandil, kui õpitegevustes on kasutusel nii auditoorse õppe kui e-õppe tegevuste elemendid; kursuse tasandil, kui auditoorsed kohtumised ja e-õppimine on osa kogu kursusest; programmi tasandil kõrgkoolis, kui kogu kraadiõpe on kombinatsioon e-kursustest ja traditsioonilistest kursustest; asutuse tasandil, kui osa kõikidest kursustest on e-kursustena. Neist enamasti on kombineeritud õppe tähenduses kasutusel kursuse tasandil kombineerimine (Graham *et al.*, 2013).



Joonis 5. Kombineerimise tasandid (Graham *et al.*, 2013).

Erinevus traditsioonilise õppe ja veebipõhise õppe vahel on vähenemas (Tarrao & Tiirmaa-Oras, 2007). Arvatakse, et lähimas tulevikus toetavad igasugust õpet digitaalsed või veebipõhised abimaterjalid rohkemal või vähimal määral. Seega on kogu õppimine asendumas kombineeritud õppimisega (Bonk & Graham, 2006).

Osguthorpe ja Graham tuvastasid kuus põhjust, millal eelistatakse kombineeritud õpet teistele meetoditele (Bonk & Graham, 2006):

1. pedagoogilised võimalused (ingl *pedagogical richness*),
2. ligipääs teadmistele (ingl *access to knowledge*),
3. sotsiaalne suhtlus (ingl *social interaction*),
4. iseseisev tulemuste saavutamine (ingl *personal agency*),
5. tasuvus (ingl *cost-effectiveness*),
6. kontrollimise lihtsus (ingl *ease-of-revision*).

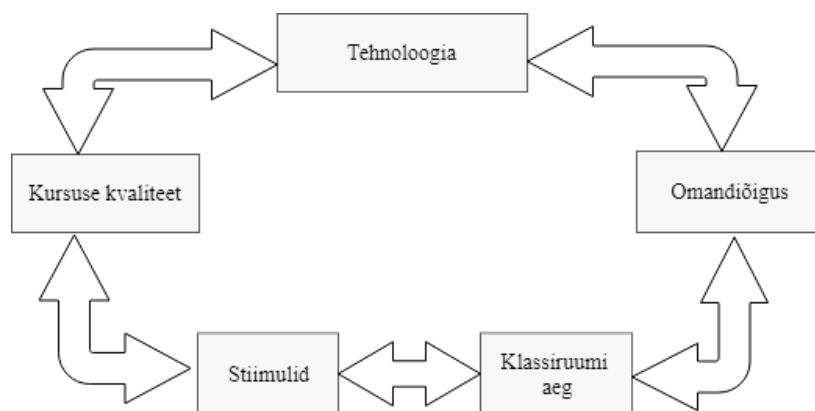
Neist kolm enim esinevat põhjust on Grahami, Woodfieldi ja Harrisoni (2013) sõnul tänu aktiivõppe kasutamisele parem pedagoogiline lähenemine, suurenenud ligipääs teadmistele ja tasuvus.

Kaur (2012) toob välja kombineeritud õppe viis eelist traditsioonilise õppe ees:

- Võimalus ümber lülituda passiivselt õppimiselt aktiivsele õppimisele varieerides õppematerjale aktiivsete tegevustega, kus õppija pannakse olukorda, kus ta peab lugema, rääkima, kuulama ja mõtlema.

- Annab õppijale võimaluse õppida kas koos kaaslastega või eraldi. Õppijatel on erinevad õppimisstiilid ning uuringute põhjal võimaldab kombineeritud õpe vähendada õppimisel ebaõnnestumiste arvu, tõsta tulemusi ja õppijate motivatsiooni.
- Õppimisele lisandub inimlik puude. Interaktiivne osa võimaldab juhendajal luua kõrgel tasemel huvi, vastutuse ja hindamise.
- Suurendab individuaalsust, personaalsust ja asjakohasust. Võimaldab juhendajal läheneda erinevatele õppijate gruppide vajadustele personaalselt.
- Õppijatel on kasutada mõlema õppe (auditoorse ja e-õppe) parimad osad, juhendajal ja õppijatel on suurem paindlikkuse võimalus ilma isiklikku kontakti kaotamata.

Kombineeritud õppe omaks võtmist ja rakendamist mõjutavad Grahami, Woodfieldi ja Harrisoni (2013) sõnul viis tegurit (joonis 6): tehnoloogia ehk vajaliku riist- ja tarkvara olemasolu, internetiühendus jms; omandiõigus ehk internetis vabalt kättesaadavate materjalide jagamisega kaasnev intellektuaalse omandi probleem; klassiruumis olemise aja asendamine veebipõhise juhendamise ajaga konkreetse kursuse jaoks; stiimulid ehk infrastruktuuri arendamine ja asutuse positiivsed seisukohad kombineeritud õppe rakendamisele, mis aitavad kaasa edukusele; kursuse kvaliteedi hindamine.

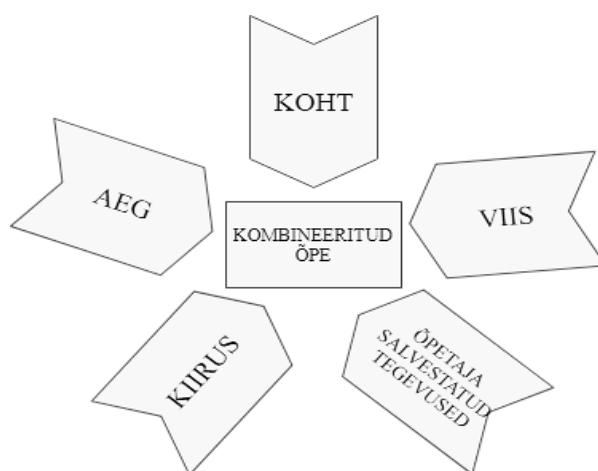


Joonis 6. Kombineeritud õpet mõjutavad tegurid (Graham *et al.*, 2013).

Uuringud on näidanud, et suurim takistus e-õppe elementide kasutusele võtmisel on puudulik tehnoloogiline infrastruktuur (Graham *et al.*, 2013). Oluliseks teguriks on samuti kombineeritud õppe kursuse kvaliteedi hindamine (Graham *et al.*, 2013).

Kombineeritud õpet iseloomustab aeg, kuna õppijad õpivad oma valitud ajal, neid ei piira õppetunni pikkus (Beaver *et al.*, 2014). Samuti õpib iga õppija omas tempos, mis tähendab, et kui vaja, kulutab rohkem aega või läheb hoopis kiiremini edasi. Õppijad õpivad nii klassiruumis kui väljaspool klassiruumi, nt kodus või raamatukogus ning õpivad kasutades

erinevaid õppimismudeleid. See võib sisaldada õppimist suurtes gruppides, meeskonnatöös väikestes gruppides, individuaalset juhendamist nii auditoorselt kui veebipõhiselt kasutades erinevaid juhendamistehnikaid (Beaver *et al.*, 2014). Beaver, Hallar ja Westmaas (2014) toovad kombineeritud õppe sisust tulenevatele aja, kiiruse, koha ja viisi olulisusele lisaks veel elemendi, et õpetaja tegevused on varasemalt salvestatud näiteks videona (ingl *teacher-of-record*) (joonis 7).



Joonis 7. Kombineeritud õppe sisu elemendid (Beaver *et al.*, 2014).

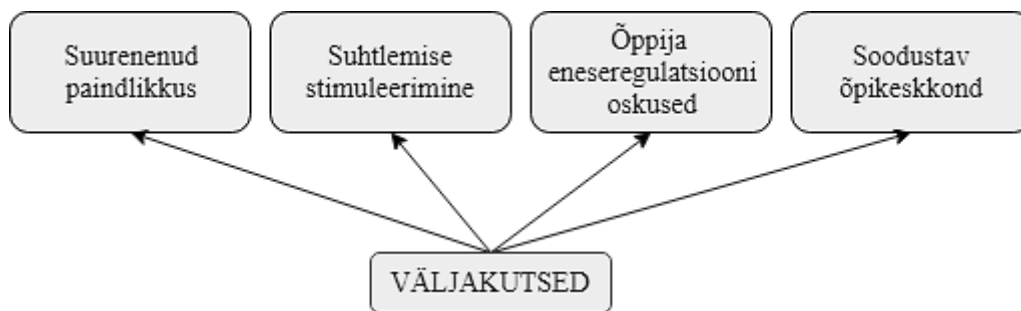
Õpetaja salvestatud tegevuste all mõtlesid Beaver, Hallar ja Westmaas õpetamist kas reaalsete või virtuaalsete õpetajate poolt, lisaks võib olla veebipõhiste õppekomponentide tugi. Erinevad kombineeritud õppe mudelid kasutavad neid viit elementi erinevates kombinatsioonides. Nad kirjeldavad kombineeritud õppes nelja soovitatavat kasutatavat mudelit (Beaver *et al.*, 2014):

- Pöörlemismudel (ingl *rotation model*). Selles mudelis vahetavad õppijad õppeviise, millest üks on veebipõhine õppimine, kas fikseeritud ajakava järgi või õpetaja suunamisel. Populaarsed alammodellid on:
 - õpijaamade pöörlemine, kui õppijad vahetavad klassiruumis õpijaamasid, millest vähemalt ühes on veebipõhine õppematerjal;
 - õppelabori pöörlemine, kus õppijad vahetavad klassiruumis õppimist ja veebipõhist õppimist, mis toimub selleks otstarbeks loodud õppelaboris;
 - ümberpööratud klassiruum, kus kindlaksmääratud ajakava järgi on osa klassiruumi õppetunde asendatud õppijate endi poolt kontrollitud veebipõhise õppimisega;
 - individuaalne pöörlemine, kus õppijad ise jaotavad oma vajadustest lähtuvalt erinevatele moodulitele kulutatavat aega.

- Paindlik mudel (ingl *flex model*). Sarnaselt individuaalsele pöörlemismudelile töötavad paindliku mudeli järgi õppijad individuaalsete ajakavade järgi vahelduvate meetoditega, millest üks on veebipõhine õppimine. Erinevus seisneb õppimise ajal õppija vajadustest lähtuvates paindlikes muutuvates ajakavades ja õppija toetamises.
- „*A la carte*” mudel. Mudel võimaldab õpilastel kujundada oma hariduskogemust, valides spetsiaalseid veebikursusi, et täiendada traditsioonilisi koolis pakutavaid kursuseid.
- Rikastatud virtuaalne mudel (ingl *enriched virtual*). Selles mudelis õpivad õppijad peamiselt veebis, kuid jagavad oma aega füüsilise koolikeskkonna ja veebikeskkondade vahel.

Kombineeritud õppe mudelid on intensiivsusest erinevad, sõltuvalt sellest, kas konkreetse kombineeritud mudeli rakendamine toob kaasa olulise muutuse klassiruumis toimuvast õppimise kogemusest. Beaver, Hallar ja Westmaas (2014) rõhutavad, et mõned kombineeritud õppemeetodid on riskantsemad ja kulukamad kui teised, kuna nad erinevad traditsioonilistest haridusmodelitest ja rakendamiseks vajalikust tehnoloogiast. Nende väitel on odavam ja väiksema riskiga õpijaamade pöörlemismudel ja labori pöörlemismudel, kuna kasutavad tehnoloogiaid, mida saab olemasolevatesse klassiruumidesse hõlpsamini integreerida. Ümberpööratud klassiruumi mudel on nende väitel kõrgema hinnaga ja suurema riskiga, kuna see mudel nõuab investeeringuid (kas õppijalt või koolilt) tehnoloogiasse, millele väljastpoolt kooliruumi ligi pääseda. Siiski rõhutavad nad, et iga mudeli riskid ja kulud varieeruvad sõltuvalt olemasolevast infrastruktuurist ning lõpliku hinnangu mudeli kvaliteedile annab mudeli rakendamisest tuleneda võiv õpilaste õppimistulemused.

Boelens, De Wever ja Voet (2017) tõid välja kombineeritud õppe kavandamise neli peamist väljakutset (joonis 8), milleks on suurenenud paindlikkus, suhtlemise stimuleerimine, õppijate eneseregulatsiooni oskused, soodsa õpikeskkonna loomine. Nad uurisid, kuidas varasemad uuringud näitavad nende väljakutsetega toime tulemist.



Joonis 8. Kombineeritud õppega kaasnevad peamised väljakutsed (Boelens *et al.*, 2017).

Kombineeritud õpikeskkonna kujundamisel on esimene väljakutse soovitava paindlikkuse lisamine, mis võib seisneda kas õppija mõningases kontrollis õppimise aja, koha, viisi või tempo üle; veebipõhise komponendi paindlikkust aja või koha osas, kuna õppijad ei pea olema ühes klassiruumis; õppija kursuse sisu läbimise järjekorra ja tempo valik; õppijate võimalus valida auditoorse õppe või e-õppe õpitegevuste vahel (Boelens *et al.*, 2017). Boelens, De Wever ja Voet (2017) leidsid kombineeritud õpet kasutavate kursuste paindlikkuses suuri erinevusi. Nad leidsid, et vähesed kursused andsid õppijale endale kontrolli kombineerimise üle ning väitsid, et põhjuseks on kursuste juhendajate arvamus muuhulgas õppijate ebapiisavast eneseregulatsioonioskusest.

Kombineeritud õpikeskkonna suurenenud paindlikkus aja ja koha osas toob ennekõike kaasa raskendatud sotsiaalse suhtlemise. Paljud õppijad soovivad pakutavat paindlikkust, kuid ei taha kaotada traditsioonilise auditoorse õppega kaasnevat sotsiaalset suhtlemist (Bonk & Graham, 2006). Paindliku aja ja kohaga õppimisel võivad õppijatel tekkinud probleemid jääda juhendajatel märkamata, mistõttu on suhtlus õppijate ja juhendajate vahel oluline (Boelens *et al.*, 2017).

Kombineeritud õpikeskkonnas on õppijate suurema paindlikkuse ja iseseisvuse tõttu muutunud oluliseks õppija eneseregulatsiooni oskused: organiseerimisoskus, distsipliin, ajajuhtimine, oskus kasutada õppimise toetamiseks tehnoloogiat ning enesekontroll (Bonk & Graham, 2006). Ennastjuhtiva õppimise suunajaks on õppija ise ning selles sisaldub omas tempos õppimine, iseseisvalt õppimine, individuaalne õppimine ja iseenda õppimise suunamine. Boelens, De Wever ja Voet (2017) toovad välja, et mitmed uuringud on näidanud, et kuna eriti kasulik on suurem paindlikkus edukatele õppijatele, kellel on enesereguleerimise oskused kõrgemad, siis kombineeritud õpe pakub väljakutset just madalamate saavutustega õppijate õppimise hõlbustamiseks.

Neljas väljakutse keskendub motiveeritud ja positiivse õpikeskkonna loomisele, millele aitavad kaasa juhendaja empaatia, huumorimeel, õppijate julgustamine, tähelepanu juhtimine ülesannete olulistele aspektidele ning õppijate individuaalsuse arvestamine, mis annab tulemusena õppijatele enesekindluse, aktsepteerituse ja väärtustatuse tunde ning soodustab positiivset hoiakut kursuse suhtes ja hõlbustab õpiprotsessi (Boelens *et al.*, 2017).

2.3.3 Kombineeritud õpe ja MOOC

MOOC-id pakuvad võimalust kombineeritud õppe kursuse kujundamiseks. Õppetööde asemel saab loenguid salvestada, pakkudes veebipõhiseid õppevideoid. Kuna MOOC-id on kellegi teise poolt kavandatud ja mõeldud läbi viimiseks eraldiseisvate kursustena, siis MOOC-i lisamine kombineeritud õppesse piirab õpetajal oma kursuse kujundamise otsuseid: veebipõhine komponent on suhteliselt fikseeritud ja ainult klassi komponenti saab muuta (Bruff *et al.*, 2013).

Koller (2012) märkis, et kuigi MOOC-e kasutatakse eraldiseisvate veebikursustena, pakuvad nad uusi võimalusi auditoorse õppe toetamiseks. MOOC-ide kasutamise traditsioonilises õppes jaotab Israel (2015) kahte kategooriasse: kui MOOC-e on kasutatud traditsioonilise õppe asemel ning kui MOOC-e on kasutatud traditsioonilise õppe toetamiseks. Üha enam teadlasi, õpetajaid ja ülikoole on hakanud traditsioonilist auditoorset õpet toetama kasutades MOOC-e (Holotescu *et al.*, 2014). Griffiths väidab, et kui MOOC-e kasutada kombineeritud õppes, võib see parandada õpilaste tulemusi ja vähendada õppekulusid (Griffiths *et al.*, 2014). Läbiviidud uuringud on näidanud MOOC-de lisamine traditsioonilistesse auditoorsesse õppesse on andnud peaaegu võrdseid või veidi paremaid õpitulemusi kui puhas auditoorse õpetamise keskkond (Holotescu *et al.*, 2014; Griffiths *et al.*, 2014). Õpilaste motivatsioon ja püsivus mängivad olulist rolli nii klassisisese kui ka veebipõhise tegevuse läbiviimisel.

Kasutades MOOC-e auditoorses õppes, tuleb lisaks arvestada, et kuigi MOOC-id on registreerumisel avatud ja tasuta, võivad MOOC-ide pakkujad piirata nende kasutamist teistes keskkondades. MOOC-ide tõhusaks kasutamiseks traditsioonilistes klassiruumis, nagu soovitasid Griffiths jt (2014), peaksid MOOC-i pakkujad tegema oma õppematerjalid moodulitena kasutatavateks ning arvestades autoriõigustega muutma nende sisu eri keskkondades kättesaadavaks. Nad peaksid tagama sisu kättesaadavuse ka tulevikus kasutamiseks.

Väljakutsed, mis on seotud MOOC-i kursuse materjalide kasutamisega, ei erine sellest, mis tekivad teise õppejõu õpiku kasutamisega. Siiski vajab MOOC-i kasutamine läbimõttlemist, et arvesse võtta MOOC-ide jaoks kättesaadavate õpikogemuste mitmekesisust ja interaktiivsust – loenguvideod, automaatkontrolliga testid, foorumid aruteludeks. Väidetakse, et MOOC-ide kasutamine kombineeritud õpetamisel võib tuua õppijatele kasu ning pakkuda õpetajatele väljakutseid oma erialaste teadmiste parandamiseks ja uute õpimudelite kasutuselevõtmise kompetentsuse ja oskuse arendamiseks (Holotescu *et al.*, 2014).

3.4.4 Kombineeritud õpe ja ümberpööratud klassiruumi mudel

Viimasel ajal on traditsiooniliste õpetajakesksete lähenemisviiside asemel pööratud suuremat tähelepanu õppijakesksetele aktiivsetele õppimisviiside kasutamisele. Üheks võimaluseks on kasutada ümberpööratud klassiruumi mudelit. Ümberpööratud klassiruumi mudelit tutvustasid Baker (2000) ning Lage, Platt ja Treglia (2000) aastal 2000 ning selle mudeli puhul on traditsiooniliselt klassiruumis toimunud tegevused vahetatud koduste õppetegevustega. Enamasti tähendab see, et õppija vaatab õppevideoid või loeb õppematerjale kodus ja lahendab ülesandeid ning arutleb kuuldu üle klassiruumis. MOOC-ide ja tavakursuste kombineerimisel on eelistatum õppevorm just ümberpööratud klassiruumi mudel. Tartu Ülikooli objektorienteeritud programmeerimise kursuse läbiviimisel on kasutatud ümberpööratud klassiruumi mudelit, kus üliõpilased lugesid vastavaid materjale ja lahendasid iseseisvalt probleemülesandeid, klassiruumis töötasid paarides, andsid tagasisidet teineteise tehtud tööle ning lahendasid lisaülesandeid (Lepp & Tõnisson, 2015).

Õppija haaramine õppeprotsessi on aktiivõppe juures kõige olulisem. Aktiivõppemeetodite kasutamine on arutusel ka programmeerimise õpetamise ja õppimise juures. Barg jt (2000) on välja töötanud probleemipõhise õppe kursused ja nad on leidnud, et suurem osa õppijatest väljendas positiivset suhtumist õppesse ja positiivset suhtumist kursusesse. O'Kelly jt (2004) leidsid, et aktiivõppe korral on õpilastel võimalus teineteist paremini tundma õppida ja mõista, kuidas teised õppijad lähenevad probleemile. Nende väitel on see õpilastele kasulik. Zhang jt (2013) tegid ettepaneku, et õppijate õpitulemuste parandamiseks võiksid õpetajad sissejuhatavate programmeerimiskursuste õpetamisel kaaluda aktiivsete õpimeetodite valimist traditsioonilise loengupõhise vormi asemel.

Bishop ja Verleger (2013) väidavad, et ümberpööratud klassiruumi mudel esindab tegelikult õppekava laiendamist, mitte pelgalt tegevuse ümberkorraldamist, kuna grupipõhine ja avatud

probleemide lahendamine lisandub tavaõppele. Samuti teatasid nad, et õpilaste suhtumine õppimisse on ümberpööratud klassiruumi meetodi korral üldiselt positiivne. Paaristöö korraldamisel on oluliseks küsimuseks paaride moodustamise viis. Üheks võimaluseks on lasta õppijatel ise paarid moodustada, kuid tavaliselt valivad õppijad sel juhul oma sõpru. Õppijad saavad paaristööst rohkem kasu, kui nad näevad, kui erinevalt inimesed mõtlevad ja lahendavad probleeme, mistõttu soovitatakse paare moodustada juhuslikult (Lepp & Tõnisson, 2015).

Ümberpööratud klassiruumi edukuses on oluline õpetaja ja õppija suhtlemine ning õpetaja märkamine, milliseid grupe on vaja rohkem juhendada ning kellele on vaja lisaselgitusi anda. Seega on õpetaja kogemus väga oluline. Aktiivõppe meetodeid kasutades on õppijad energilisemad ning õpetaja muutub infoandjast tagasiside andjaks (Vaikjärv & Pilli, 2015). Üheks peamiseks ohuks ümberpööratud klassiruumi meetodi puhul on, et õppija ei tee oma kodust tööd enne auditoorset tundi ning seetõttu ei tööta tunnis aktiivselt kaasa. Selle vältimiseks ja õppijate motivatsiooni hoidmiseks soovitatakse jälgida, et iseseisva töö maht ei oleks liiga suur ning samuti on võimalus kasutada kontrollülesande andmist (Vaikjärv & Pilli, 2015). Õpetaja ülesanne on sekkuda ka juhul, kui õppijal on tekkinud mahajäämus puudumise tõttu. Ümberpööratud klassiruumi meetodi kasutuselevõtule kulub õpetajal palju ettevalmistusaega, kuid kasutades juba olemasoleva kursuse materjale, on see ajakulu väiksem, kuna õppijatele tuleb anda vaid olemasolevate õppematerjalide lingid. Lisaks teadmiste andmisega õppijale, annab see meetod võimaluse arendada ka õppijate ajajuhtimis-, meeskonnatöö- ning analüüsioskusi (Vaikjärv & Pilli, 2015).

3. Metoodika

Metoodika peatükis antakse ülevaade õpilaste programmeerimisalase tausta uurimisest, õppematerjalide kaardistamisest, õpilaste kursusel edasijõudmise analüüsimisest ning analüüsile tuginedes koolidele programmeerimise valikkursuse ainekava koostamisest.

3.1 Taust

Töö eesmärgi saavutamiseks viidi 2018. aasta jaanuarist aprillini läbi programmeerimise algõppe kursus kombineeritud õppena ühes Eesti väikelinna üldhariduskoolis 9. ja 10. klassi õpilastele. Koolis õpib ligikaudu 500 õpilast, neist 9. klassides kokku 44 õpilast ja 10. klassis 27 õpilast. Programmeerimise valikkursusi põhikoolis ega gümnaasiumiastmes ei õpetatud kuni 2016/17. õppeaastani. Töö autori ettepanekul koostati gümnaasiumiastme 10. klassile uus valikkursus „Programmeerimise alused”, kestusega 35 akadeemilist tundi 2016/17. õa, sel aastal viidi kursus läbi uuendatud kujul. Kursuse alguseks valiti jaanuar, kuna septembris alustamine ei sobinud töö autorile suure töökoormuse tõttu.

Algselt 10. klassile mõeldud valikkursusele oli programmeerimise huvilisi valikkursuse avamiseks liiga vähe, vaid kuus õpilast, mistõttu pakuti võimalust kursusel osalemiseks ka sama kooli 9. klasside õpilastele. Kokku registreerus kursusele kuus 10. klassi õpilast ja kümme 9. klasside õpilast, kokku 16 õpilast, neist 4 tütarlast ja 12 noormeest, kokku kolmest erinevast klassist. Pärast esimest tundi loobus kohe kursusest üks 9. klassi noormees, kes väitis, et programmeerimine võtab liiga kaua aega, mistõttu jäi kursusele 15 õpilast. Kõikide kursusel osalevate õpilaste vanemaid teavitati kursuse andmete kasutamisest magistritöös ning küsiti kirjalik nõusolek õpilaste isikuandmete kasutamiseks. Õpilaste hulgas oli kolm matemaatikaolümpiaadil osalenud õpilast ning kaks matemaatikas õpiabi saavat õpilast. Kõigil õpilastel oli kodus kasutada arvuti.

Kursuse läbiviimisel kasutati Tartu Ülikooli arvutiteaduse instituudi poolt loodud õppematerjale, mida oli eelnevalt kasutatud õpilaste MOOC-i „Programmeerimise alused” läbiviimisel. Kursuse jaoks tehti Courses-keskkonnas vastava kursuse materjalidest koopia ning töö autorile anti materjalide muutmise ja kohandamise õigused. Eelnevalt oli töö autor osalenud ka 2017. aasta oktoobrist detsembrini vastava MOOC-i korraldamisel, valikkursuse läbiviimise hetkel MOOC-i läbi ei viidud. Lisaks kasutati kursuse läbiviimisel vastava MOOC-i Tartu Ülikooli Moodle'i õppekeskkonnas oleva kursuse koopiat, milles olid antud

töö autorile õppejõu roll ning kursuse administreerimise õigused. Õpilastele tehti Tartu Ülikooli Moodle'i keskkonda ligipääsuks kasutajakontod ning lisati nad kursusele õppijate rollis. Kursuse Courses-keskkonna lehel asuvad õppematerjalid olid õppijatele kättesaadavad Moodle'i keskkonnas olevate linkide kaudu. Kursusel kasutati õppimiseks programmeerimiskeskonda Thonny, mis laeti õpilaste poolt kooli arvutiklassi arvutitesse ning paluti neil ka oma koduarvutitesse laadida.

Kursuse eeltutvustamiseks, vanemate nõusolekute lehtede jagamiseks ning avaküsitluse läbiviimiseks toimus 15. jaanuaril üks auditoorne tund, kus programmeerimise teemadega ei tegeletud. Sisulised õppetunnid viidi läbi alates 22. jaanuarist kuni 17. aprillini, kokku 12 nädalat õppetööd. Peale 5. nädalat oli üks nädal koolivaheaega ning peale seda veel 7 nädalat õppetööd. Auditoorsed tunnid toimusid paaristundidena esmaspäeviti, 7. ja 8. tund, välja arvatud viimasel nädalal, mil tunnid toimusid teisipäeval 7. ja 8. tunnina. Programmeerimise auditoorsed tunnid lisandusid õpilaste senisele nädalakoormusele, samal ajal ei toimunud teisi valikkursuseid, mis oli ka töö autori arvates üheks põhjuseks, miks õpilaste huvi kursusel osalemise vastu jaanuariks rauges võrreldes septembriga. Murede korral paluti õpilastel mitte kirjutada materjalides antud aadressile, vaid pöörduda oma kursuse õpetaja poole. Iga nädal anti õpilastele info materjali kohta, mis tuli neil enne järgmise nädala auditoorset tundi läbi lugeda, planeeritud oli selleks õpilastel kulutada kuni 45 minutit. Erinevalt MOOC-ist ei olnud Moodle'i keskkonnas avatud foorumeid. Kursuse läbimise kohta avati päevik kooli e-päevikus, kus kirjutati lahti tundide tegevused, märgiti puudujad ning anti õpilaste iseseisva lugemise teemad koduse tööna. Kirjalik suhtlus õpilastega käis vajadusel e-päeviku kaudu sõnumeid saates. Sama suhtluskanalit kasutati ka õpilastele tagasiside andmiseks ning mahajäämuse likvideerimise ergutamiseks.

3.2 Ainekava arendus

Kursuse läbiviimise aluseks oleva MOOC-i materjalid on jaotatud kaheksaks nädalaks: sissejuhatus, tingimuslause, tsükkel, sõned ja graafika, järjend, funktsioon, andmevahetus ja lihtne kasutajaliides ning lõpuks kordamine. Kursuse mahuks on 3 EAP ehk 78 tundi. Materjalid on avalikult kättesaadavad Courses-keskkonna lehel ning kursuse ametlik osa toimub Moodle'i õppekeskkonnas, kus on teemade jaotus Courses-keskkonna lehel oleva jaotusega sama. Moodle'is tuleb kursusel osalejal esitada kohustuslikud programmeerimise

ülesanded ja lahendada testid. Lisaks on võimalus suhelda nii kursuse korraldajate kui teiste kursusel osalejatega foorumis.

Koolis on ühe kursuse pikkus 35 akadeemilist tundi, millega piirduti ka töö autori poolt tundide läbiviimisel. Kuna 35 tundi sisaldas ka õpilaste iseseisvaid e-tunde, siis võrreldes MOOC-i kursuse arvestuslike tundide mahuga (3 EAP ehk 78 tundi), oli see mõnevõrra vähem. Kuna kombineeritud õppe läbiviimisel on uuringute põhjal seni kasutatud hinnanguliselt 30% tundide läbiviimist e-õppena, siis kuna arvestuslikult on 30% 35 tunnist ligikaudu 11 tundi, jaotas töö autor kursuse 35 tundi 11 iseseisvaks e-õppe tunniks ning 24 auditoorseks tunniks. Tunnid jaotati kokku 12 nädalale arvestusega, et esimesel nädala e-õppe tundi ei toimu. Kursuse algus valiti jaanuaris nii, et kursus lõppeks enne aprillis algavat koolivaheaega ning veebruaris olev koolivaheaeg jääks kursuse kesksaia ligidale.

Selleks, et ka kursuse sisulist mahtu õppija jaoks ühes nädalas vähendada, otsustati MOOC-i kaheksa nädala teemad jaotada 11 nädala peale. Jaotamisel jälgiti, et teemad oleksid loogiliselt jaotatud ning igal nädalal ei oleks rohkem kui neli kohustuslikku esitatavat programmeerimisülesannet ning võeti vastu otsus jooksvalt teemade järjestust muuta või ümber tõsta, kui selleks tekib vajadus (plaan ja selle realiseerumine on toodud töö I lisas). MOOC-i materjalide teema „Silmaring” tekstid anti soovitatava loetava tekstina, kuna mõned testide küsimused olid nende põhjal koostatud, aga teema „Alkeemia” lugusid kursusele ei võetud ning need teemad pandi ka Courses-keskkonna lehel õpilaste eest peitu (siiski linkidel „järgmine” klõpsates oli võimalik neid lugusid näha).

Iga nädal jälgiti õpilaste edasijõudmist ning viidi läbi tagasiside küsitlused ning vastavalt sellele kohandati jooksvalt tehtud nädalate jaotuse plaani. Kohe algul arvestati võimalusega, et kui mõni teema osutub õpilastele liiga raskeks või kui e-õppe osale kulub enamikul õpilastest planeeritust rohkem aega, siis võetakse mõni raskem programmeerimisülesanne kohustuslike esitatavate ülesannete plaanist ära või tehakse vabatahtlikult lahendatavaks. Nii tehti MOOC-i seitsmenda nädala teemaga. Toimunud tundide tegeliku jaotuse järgi ja tehtud ülesannete põhjal koostati programmeerimise aluste õpetamiseks programmeerimiskeeles Python ainekava (toodud töö II lisas) ning üksikasjalik tööplaan (toodud töö III lisas), õpetajale kasutamiseks.

3.3 Kursuse uurimine

Kursuse sisu oli jaotatud 11 nädalaks, viimaseks nädalaks planeeriti kursuse lõpetamiseks arvestustöö kirjutamine. Courses-keskkonna lehel asendati esimese nädala materjalide tekst programmiga H5P tehtud slaidiesitlustega (kus oli varasem materjalide tekst), kuhu lisati kontrollküsimusi. Teistel nädalatel Courses-keskkonna lehe materjalide hulgas muutusi ei tehtud töö autori ajapuudusel. Moodle'i nädalate jaotust muudeti jooksvalt vastavalt tehtavatele muutustele.

Esimesel kohtumisel viidi läbi avaküsitlus (küsimused on toodud töö IV lisas), et teada saada õpilaste varasemaid kokkupuuteid programmeerimisega. Igal nädalal koguti tagasisidena õppijate hinnanguid uute teemade ja lahendatavate ülesannete raskuse kohta, et saada infot õppimise tempo sobivusest õpilastele ning olla kiirelt valmis muutusi läbi viima. Nii avaküsitlus kui tagasiside küsimused tehti veebipõhise küsitlusena kasutades Google Drive'i.

Iga uue nädala materjalid tehti õpilastele kättesaadavaks vastava nädala esmaspäeva hommikul. Nädalate kaupa olid Moodle'is lingid õppematerjalidele ja ülesannetele. Programmeerimise ülesandeid kontrolliti Moodle'is automaatselt ning kui esitatud programm vastas nõutud tingimustele, hinnati lahendust hindegas „arvestatud”. Nädalatel 2, 4, 5, 6, 7, 10 ja 11 olid lahendamiseks ka testid, milles oli tarvis õigesti vastata vähemalt 90% küsimustest, siis oli hindegas samuti „arvestatud”. Nii ülesandeid kui teste sai esitada mitu korda. Moodle'is olevad hindamise arvestus andis iga nädala kokkuvõtvaks hindegas „arvestatud”, kui kõik vastava nädala kohustuslikud ülesanded olid saanud hindegas „arvestatud”. MOOC-i kursuse hindamise arvestus tehti ümber nii, et need sobiks läbiviidava kursuse ajalise kestusega ning kohustuslike ülesannetega. Kursusel oli kasutusel eristav hindamine.

Iga nädal märgiti e-päevikusse kolm ainetundi (kaks auditoorset tundi ja üks iseseisev e-õppe tund) esimesel nädalal kaks tundi (mõlemad auditoorsed tunnid). Igal nädalal toimunud kahest auditoorsest tunnist esimesel toimus paaristöö iseseisvalt loetud teema materjali põhjal koostatud ülesannetega (iga nädala ülesannete lingid on kättesaadavad töökavas, mis on toodud töö III lisana), mis kinnistasid loetud uut teadmist. Paare võisid õppijad esimesel kolmel nädalal ise moodustada, järjestikutel nädalatel paluti neil leida paariline, kellega eelneval nädalal koos ei töötatud. Edaspidi moodustati paare juhuslikkuse alusel. Kui kohal olevate õppijate arv oli paaritu, anti õpilastel valida või otsustas õpetaja, milline rühm koosnes kolmest õppijast.

Kursuse viimasel nädalal pidi õppija sooritama omal valikul kas arvestustöö ülesande või tegema loovtöö, millest teavitati õppijaid 3 nädalat ette. Arvestustööks valiti MOOC-i kursuse arvestustöö ülesanne. Aega oli ülesande sooritamiseks kaks akadeemilist tundi ning ülesandel oli Moodle'is automaatkontroll, mis hindas õige lahenduse hindegga „arvestatud”, mis õppija e-päevikus esitati kui hinne „5”. Õpilastele, kellel jäid ülesande lahendusse vead sisse, anti kõigepealt vihjeid lahenduse parandamiseks ning hiljem iseseisvaks lahendamiseks lisaülesanne. Loovtööle esitatavad nõuded olid samuti samad mis MOOC-i kursusel, kuid kasutamata võis jääda kolm tingimust (MOOC-i kursusel kaks tingimust). Loovtööd tuli õppijatel üles laadida Moodle'i foorumis, kus õppija nägi teiste loovtöid alles peale enda loovtöö esitamist. Loovtöid hinnati Moodle'is hindegga „arvestatud” ning õpilase e-päevikus hindegga „5”, „4” või „3” sõltuvalt sellest, kas õppija vajas oma töö lõpetamiseks õpetaja vihjeid või abi.

Kursus lõppes Google Drive'is läbi viidud lõpuküsitlusega (küsimused on toodud töö V lisas) peale viimasel nädalal toimunud arvestustöö või loovtöö sooritamist, et saada tagasisidet kogu kursuse kohta. Küsimustiku koostamisel võeti aluseks MOOC-il läbi viidud lõpuküsitlus ning kohandati sealseid küsimusi töö eesmärgist lähtuvalt. Küsimustena kasutati valikküsimusi, lühivastusega küsimusi ja küsimusi, milles õppijate hinnangud olid viiepallises skaalas 1-st 5-ni: 1–ei nõustu, kuni 5–nõustun.

Andmed küsitlustest ning saadud andmed ülesannete esitamiste kohta Moodle'ist koguti Google Drive'i, tehti statistiline andmeanalüüs kasutades programmi Microsoft Excel 2013 ning esitati saadud tulemused tabelitena ning diagrammidena.

4. Tulemused ja arutelu

Selles peatükis antakse ülevaade koostatud ja läbiviidud programmeerimise algõppe kursuse „Programmeerimise alused” materjalide ja ainekava kujunemisest. Samuti analüüsitakse kursuse õppijate edukust ning arvamusi kursuse kohta ning tehakse neile tuginevalt järeldusi ning ettepanekuid kursuse parendamiseks.

4.1 Ainekava ja kursuse materjalide koostamine

Järgnevalt kirjeldatakse, kuidas koostati ainekava, milliseid teemasid käsitleti ning millised olid vastavate nädalate kohustuslikud tegevused nii auditoorsetes tundides kui e-õppena. Samuti antakse ülevaade õppijate hinnanguid nädala teemade mõistmise kohta ning tuuakse näiteid tunnis paaristöödena kasutatud ülesannetest.

4.1.1 Ainekava

Kursuse kasutatavaks õppematerjaliks oli vastava MOOC-i „Programmeerimise alused õpilastele” õppematerjalid, mis on kättesaadavad aadressil <https://courses.cs.ut.ee/2018/blprogal/spring> ning kus on teemad jaotatud kaheksaks nädalaks. Kogu kursuse arvestuslik maht on 78 tundi keskmise nädala ajakuluga 8–9 tundi. Lisaks on MOOC-i kursusel erinevad silmaringi avardavad tekstid ning läbi nädalate jooksev kuuest osast koosnev lugu „Alkeemia”.

Valikkursuse „Programmeerimise alused” koolis kombineeritud õppe kursusena kestuseks oli planeeritud kokku 35 akadeemilist tundi (iga tund 45 minutit) keskmise nädalakuluga 3–4 tundi. Auditoorse õppena viidi läbi 24 tundi (iga nädal üks paaristund, kokku 12 paaristundi) ja iseseisva e-õppena 11 tundi. Kuna kursuse maht 35 tundi on ligi poole väiksem MOOC-i planeeritud ajakulust, jäeti kursuse kohustuslike teemade hulgast välja MOOC-i „Alkeemia” teemad. „Silmaringi” teemad anti siiski õppijatele lugemiseks, kuna neist materjalidest said õppija huvitavat lisainfot ning mõned testide küsimused toetusid neile materjalidele.

Esialgu oli planeeritud läbida kõik MOOC-i teemad vähendades kohustuslike esitatavate programmeerimisülesannete arvu. Kursust läbi viies jälgiti õppijate edenemist ning korrigeeriti vastavalt plaani ning vähendati läbitavate teemade arvu. Kursuse teemade läbimise plaan ja tegelikult väljakujunenud kursuse nädalate teemad on toodud töö lisa (I lisa).

Kursuse lõpetamisel saavutatavad õpitulemused on tuletatud MOOC-i kursuse saavutatavatest õpitulemustest (Programmeerimise alused õpilastele, s.a.):

- õppija oskab programmeerimise baaskonstruksioone (tingimuslause, tsükkel, funktsioon) esitada nii plokk skeemidena kui ka programmilõikudena programmeerimiskeeles Python;
- õppija oskab lihtsa ülesande realiseerida programmina programmeerimiskeeles Python, kasutades seejuures erinevaid baaskonstruksioone ja andmetüüpe (täisarvud, ujukomaarvud, sõned, järjendid), muutujaid, operatsioone, graafikavahendeid ning andmete lugemist failist;
- õppija oskab leida ja sõnastada enda eri- või huvialaga seotud programmeerimisülesandeid ja neid lahendada;
- õppija omab esmast ülevaadet programmeerimise ajaloost, erinevatest programmeerimiskeeltest, arvusüsteemidest ja regulaaravaldisest;
- õppija on mõelnud läbi, kas ta tahab end edaspidi programmeerimisega rohkem kurssi viia ning on teadlik sellekohastest võimalustest.

Kursuse algusele eelneval nädalal viidi läbi kursust tutvustav tund, kus teavitati õppijaid oodatavatest õpitulemustest, kursuse ülesehitusest, hindamissüsteemist ning viidi läbi avaküsitlus õppijate programmeerimisalase taustaga tutvumiseks. Kuna kõik kursusel osalejad olid ilma eelneva programmeerimiskeele Python kogemuseta ning keegi neist ei olnud õppinud ka süsteemselt ühtegi teist programmeerimiskeelt, koostati kursuse töökava nii, et see oleks sobilik ka algajale programmeerimishuvilisele õppijale.

Kursuse sisu jaotati 12 nädala peale, neist esimesed 10 nädalat tegeleti erinevate programmeerimise teemadega, 11. nädalal korraldati kõike õpitut ning 12. nädalal toimus kursuse lõputöö, milleks õppija võis valida arvestustöö või loovtöö. Kursuse jooksul ei jõutud läbida MOOC-i seitsmenda nädala teemat „Andmevahetus. Lihtne kasutajaliides”, mis jäi kursuse kohustuslike teemade hulgast välja. Siiski anti vastava teema õppimise võimalus lisateemana kolmele kiiremale õppijale ning valmistati neile ette vastavad auditoorse tunni ülesanded ning avati Moodle’is vastav teema. MOOC-i nädalate teemade jaotus ja kursuse nädalate teemade võrdlus on toodud tabelis 1.

Tabel 1. MOOC-i ja kombineeritud õppe kursuse teemade läbimise ajakava võrdlus.

Nädal	Teema MOOC-is	Teema kombineeritud õppes
1.	Sissejuhatus	Sissejuhatus
2.	Tingimuslause	Sissejuhatus
3.	Tsükel	Tingimuslause
4.	Sõned. Graafika	Tingimuslause
5.	Järjend	Tsükel
6.	Funktsioon	Sõned. Graafika
7.	Andmevahetus. Lihtne kasutajaliides	Järjend
8.	Lõpuks	Järjend
9.	– (MOOC on lõppenud)	Funktsioon
10.	– (MOOC on lõppenud)	Funktsioon
11.	– (MOOC on lõppenud)	Lõpuks
12.	– (MOOC on lõppenud)	Lõpuks

Kuna enne esimest tundi ei olnud õppijatele antud eelnevat ettevalmistavat tööd e-õppena, siis oli planeeritud esimese nädala auditoorsetesse tundidesse õpetaja selgitused, e-materjalide lugemine ja sealsete slaidide uurimine ning enesekontrolli testide lahendamine. Iga nädala paaristunni lõpus hindasid õppijad oma uuest teemast arusaamist skaalal 1-st kuni 5-ni, kus väärtus 1 tähistas, et teema oli õppija jaoks kerge ning väärtus 5 tähistas, et teema oli õppija jaoks raske. Õppijate nädalate 1 kuni 10 hinnangute ülevaade on toodud tabelis 2. Kuna 11. nädalal enam uut teemat ei õpitud, siis ei küsitud ka 11. nädalal õppijate hinnanguid teema arusaadavuse kohta.

Tabel 2. Õppijate hinnang nädala teemade arusaadavuse kohta.

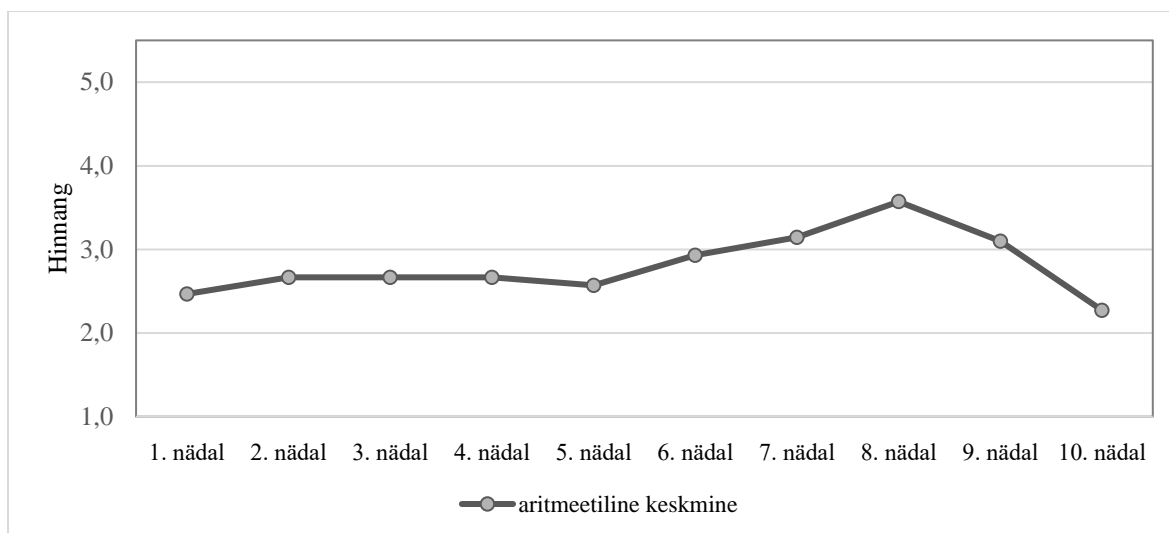
Nädal	Alateemad	Min	Max	Keskmine	Standardhälve
1.	Algoritm. Programm. Andmetüübid.	1	3	2,47	0,64
2.	Andmetüübid. Muutujad. Sisend kasutajalt: <i>input()</i> .	1	4	2,67	0,72
3.	Tingimuslause. Tingimuslause tingimuslause sees. Mitmeosaline tingimus. Loogilised tehted ja avaldised. Mitmeharuline	2	4	2,67	0,72

Nädal	Alateemad	Min	Max	Keskmine	Standardhälve
	tingimuslause <i>elif</i> abil.				
4.	Juhuslik arv. Kilpkonnagraafika.	2	4	2,67	0,62
5.	Tsükkel. Tsükkel - igal sammul juhtub midagi. Tsükkel - mitu korda? Juhuslikust arvust veel.	1	4	2,57	0,85
6.	Sõned. Graafikaraamistik <i>tkinter</i> . Tahvel. Kasutajaliides <i>tkinteriga</i> .	2	4	2,93	0,92
7.	Järjend. <i>For</i> -tsükkel, funktsioon <i>range</i> .	2	5	3,14	1,03
8.	Andmed failist.	2	5	3,57	0,85
9.	Funktsioon. Funktsioonil on väärtus.	2	4	3,10	0,88
10.	Argumendid, muutujad. Graafika ja funktsioon.	1	3	2,27	0,65

Õppijate hinnangute põhjal saab väita (keskmine 2,47; standardhälve 0,64), et esimene kokkupuude programmeerimiskeelega Python oli õppijatele sobiva arusaadavusega. Alates kolmandast nädalast avati Moodle'i keskkonnas õppijate palvel ülesannete esitamiste võimalus ka peale esitamise esialgset lõpukuupäeva, et õppijad saaksid ülesandeid kodus lõpetada. Peale viienda nädala tunde oli koolis nädalane vaheaeg.

Kõige raskemaks on hinnatud kaheksanda nädala teemat „Andmed failist” (keskmine 3,57; standardhälve 0,85). Kuna peale viiendat nädalat oli olnud koolis vaheaeg, siis töö autori arvates oli ka see seotud õppijate hinnangutega, kuna vaheajal võis osa seni õpitust ka ununeda. Seitsmenda nädala hinnangud andsid töö autorile märku, et õppijate jaoks on tarvis õppimise tempot aeglasemaks võtta, et nad jõuaksid teemadesse süveneda (keskmine 3,14; standardhälve 1,03). Seega kaheksanda nädalale planeeriti vaid üks uus teema ja auditoorsetes tundides tegeleti ka eelneva nädala ülesannetega. Kümnenda nädala hinnangutele tuginedes saab väita, et õppimise tempo aeglustamine õigustas (keskmine 2,27; standardhälve 0,65).

Seega kokkuvõttes saab väita (joonis 9), et esimese viie nädala teemad olid õppijatele arusaadavad ning teemade läbimise tempo sobis õppijatele. Raskused tekkisid peale koolivaheajaga, kuuendal nädalal ning enam raskust valmistasid 7. ja 8. nädala teemad.



Joonis 9. Teemade arusaadavus, 1–väga kerge, ..., 5–väga raske.

Üheteistkümnenda nädala mõlemas auditoorses tunnis lahendati kordavaid paaristöö ülesandeid. Kordamise teemad olid õppijate endi poolt välja pakutud eelneva nädala tagasisides. Kaheteistkümnendal nädalal viidi läbi lõputöö, milleks õppija võis valida arvestustöö ja loovtöö vahel. Arvestustööks oli MOOC-i kursuse arvestustöö ülesanne, loovtöö jaoks vähendati töös kohustuslikke kasutatud elementide nõudeid võrreldes MOOC-i loovtöö nõuetega. Õppijad said loovtöö nõuded teada loovtöö kirjutamise tunni algul.

Kuna läbiviidud kursuse üheks lisaeesmärgiks oli suurendada ja hoida õppijate huvi programmeerimise vastu, siis lähtuti hindamissüsteemi koostamisel kaasaegsest õpikäsitlusest ning õppijat arengut toetava hindamise põhimõtetest ning anti edasijõudnutele lisateemasid ning -ülesandeid. Tagasisidet said õppijad tehtavate paaristööde kohta nii õpetajalt kui ka paariliselt suuliselt, samuti digitaalsete vahendite poolt (Thonny's programmi kompileerumine, veateated, automaatkontrolliga Moodle'i testid ja programmeerimis-ülesanded).

Moodle'is oli hindamine seatud nii, et iga nädala kohustuslike ülesannete eest tuli õppijal saada „arvestatud”, mis kokkuvõttes andis nädala hindeks „arvestatud” ning kogu kursuse kokkuvõtteks samuti „arvestatud”. Kuna valikkursus oli eristava hindamisega, siis õppija nädala kokkuvõttev hinne „arvestatud” sisestati kooli e-päevikus kui hinne „5”. Kui õppija puudus või oli mõnel muul põhjusel maha jäänud, anti võimalus konsultatsioonitunniks. Kui õppija ei olnud kasutanud võimalust abile ning mahajäämus oli ühe nädala teemaga, märgiti see e-päevikus tähisega „!”, mis on koolis kokkulepitud tegemata töö märg. Kui mahajäämus oli 2 nädalat, muutus tähis „!” hindeks „1”, kui vastaval nädalal ei olnud

sooritatud mitte ühtegi nõutud ülesannet, hindeks „4” või „3”, kui vastava nädala nõutud ülesannetest oli sooritatud vastavalt rohkem kui 75% või 50%. Samas anti õpilasele võimalus saadud hinnet „1” või „3” parandada kahe nädala jooksul maksimaalselt hindeks „4”. Õppijaid ergutati kõiki ülesanded ise lõpetama ning kõikide Moodle’i nädalate eest „arvestatud” saama. Kohustuslikeks teemade arvuks kujunes kursuse jooksul üheksa, mistõttu oli igal õppijal e-päevikus kursuse lõpuks üheksa nädala hinnet ning lõputöö hinne. Kursuse kokkuvõttev hinne kujunes saadud nädalate hinnete aritmeetilise keskmise ja arvestustöö hinne põhjal.

4.1.2 E-õpe

Kursust toetavaks keskkonnaks oli kasutusel Tartu Ülikooli Moodle’i keskkond, kus oli toodud kursuse sisu jaotus nädalate kaupa. Kursuse 35-st tunnist 11 tundi viidi läbi e-õppena kasutades ümberpööratud klassiruumi mudelit. Selleks anti õppijatele iga nädal e-tunnina iseseisvaks valmistumiseks järgneva nädala teemad. Vastav iseseisev töö ja viited loetavale materjalile oli esitatud Moodle’i keskkonnas. Materjalid sisaldasid enesekontrolli küsimusi ning selgitavaid videoid. Iganädalastele tagasiside vastustele tuginedes kulus õppijatel iseseisvale e-õppele igal nädalal keskmiselt 42 minutit. Lõpuküsitluses küsiti samuti õppijate hinnangut iseseisva aja läbimisele, selle tulemus oli keskmiselt 41 minutit. Üks õppija ei kulutanud mitte ühelgi nädalal kodus uue materjali lugemisele aega, kuna jõudis juba paaristunni jooksul nii kõik antud kohustuslikud kui ka lisaülesanded tehtud. Kui mitte arvestada keskmise hulka seda ühte õppijat, kes enda väitel ei kulutanud iseseisvale tööle kodus üldse aega, siis oli õppijate keskmine iseseisva töö aeg nende endi hinnangul 44 minutit. Töö autori arvates on see väga hea tulemus, kuna sama palju aega oli selleks tegevuseks ka planeeritud.

E-õppe tegevuse kohta eraldi arvestust ei peetud ning seda ka ei kontrollitud, kas õppija ikka tegeles e-õppega. Igal õppijal oli endal vastutus oma õppimise eest. Kaudselt kontrolliti e-õppe tegevust iganädalases paaristöös, kus eelnev valmistumine andis paaristöö tegemisel eelise. Samuti avaldasid õppijad tagasisides arvamust, et tunniks mittevalmistumine pigem raskendab teemast arusaamist. Kuna läbiviidud kursus oli õppijatele valikkursus ja kõik kursusel õppijad väitsid avaküsitluses, et tahtsid ise sellele kursusele tulla, siis võis töö autor eeldada, et kursusele tulid õppijad, kes tahtsid programmeerimist õppida ning neid ei pea selleks sundima.

4.1.3 Tegevused auditoorses tunnis

Iga nädal toimunud paaristunni esimeses tunnis tegid õppijad paaristöö ülesandeid. Ideed paaristöö ülesannete koostamiseks sündisid koostöös juhendajatega, kellel on varasemaid kokkupuuteid programmeerimise õpetamisel paaristöö läbiviimise ning uurimisega. Ülesanded olid koostatud töö autori poolt vastavaks tunniks iseseisvalt loetud materjalist arusaamise toetamiseks ning tulevaste kohustuslike programmeerimisülesannete lahendamiseks ettevalmistamiseks. Nädalate kaupa paaristöö ülesanded erinesid vaheldusrikkuse ja huvi hoidmise eesmärgil, samuti olid õppijatel erinevatel nädalatel üldjuhul erinevad paarilised. Kokkuvõtte erinevatest ülesannetest on toodud tabelis 3, viited kõikidele koostatud ülesannetele on lisatud koostatud töökavasse, mis on töö lisas (III lisa). Paaristöö toimumise ajal liikus õpetaja paaride juures ning jälgis toimuvat, vajadusel esitas suunavaid lisaküsimusi või täpsustavat infot, vastas õpilaste küsimustele või esitas ise õpilastele lisaküsimusi, kui märkas selle vajalikkust. Paaristunni teises tunnis tegelesid õppijad üldjuhul Moodle'is olevate kohustuslike programmeerimisülesannete ja testide lahendamisega.

Tabel 3. Paaristöö ülesannete näited ja kasutuse aeg.

Kasutus	Paaristöö ülesannete tekstide näited
Nädal 1	Ülesanne. Arva, mis tehtega on tegemist ja mis väljastatakse ekraanile. Kontrolli oma arvamust Thonnyga. 1) <code>print(5/2)</code> 2) <code>print(5//2)</code> 3) <code>print(str(1.7)+str(1.7))</code>
Nädal 2	Ülesanne. Risto tahtis kirjutada programmi, mis antud ruudu übermõõdu järgi arvutaks ruudu pindala. Leia vead programmis ja paranda need. Programm: <pre>P = input("Sisesta ruudu übermõõdt: ") kül = P / 4 S = kül ** 2 print(Ruudu pindala on S)</pre>
Nädal 3	Ülesanne. Antud on töötav programm. Mida teeb programmi iga rida? Kontrolli Thonnyga. Programm: <pre>arv = int(input("Sisesta arv: ")) if arv % 2 != 0: arv += arv print(arv)</pre>

Kasutus	Paaristöö ülesannete tekstide näited
	<pre>else: arv -= 2 print(arv)</pre>
Nädal 4 Nädal 8 Nädal 9	Ülesanne (paarisprogrammeerimine). Kirjutage programm, mis mängib kasutajaga arvu äraarvamise mängu. Näiteks küsib programmi kasutajalt ühe arvu etteantud vahemikus, näiteks 5-st 10-ni ning ütleb seepeale, kas kasutaja arvas ära õige arvu.
Nädal 5 Nädal 10	Ülesanne. Antud on töötav programm. Arvake, mille programm väljastab ekraanile. Kontrollige oma arvamust Thonnyga. Programm: <pre>i = 0 summa = 0 while summa < 5: summa += 1 i +=1 print(i)</pre>
Nädal 7	Ülesanne. Programmis on antud selline järjend: <pre>a = [1, 12, -3, 4, 0]</pre> Milline rida lisada programmi, et ekraanile trükitaks selline järjend, kus esimest ja viimast arvu ei ole? Ekraanile prinditakse: <pre>[12, -3, 4]</pre> Kontrollige oma arvamust Thonnyga.
Nädal 11	Ülesanne. Programmis on read sassi aetud. Tõstke read sobivatesse kohtadesse nii, et programm töötaks. Mida programm teeb? Kontrollige oma arvamust Thonnyga. Programmi read: <pre>fail.close() nimed = [] print("Tänane menüü: " + str(nimed[k-1])) nimed.append(rida) k = int(datetime.now().day) for rida in fail: from datetime import * fail = open(nimi, encoding = "UTF-8 ") nimi = str(input("Sisestage failinimi: "))</pre>

Esimesel nädalal, kui õppijatel ei olnud eelnevalt iseseivat õppimist olnud, tegeleti paaristöö asemel töölehe järgi erinevate andmetüüpide arusaamise õppimisega. Samuti harjusid õppijad programmis matemaatiliste tehete senisest neile tuttavast kirjaviisist erineva üleskirjutusega.

Teisel nädalal olid paaristöö ülesanneteks vigadega programmid, kus tuli teha parandusi. Paarilised pidid arutama ülesandeid omavahel ja kõigepealt püüdma parandused teha paberil. Hiljem katsetati enda paberil kirjutatud lahendusi Thonnyga. Kolmandal, viiendal ja kuuendal nädalal olid ülesanneteks töötavad programmid, mille puhul tuli aru saada, mis on programmi töö tulemus või mis ülesanne on igal programmi real. Lisaks oli antud ülesanne, kus paarilised pidid etteantud plokkiskeemi järgi koostama programmi. Neljandal, kaheksandal, üheksandal ja kümnendal nädalal lahendati paarisprogrammeerimise ülesanded. Paarisülesannete puhul oli algul õppijatel raskusi graafikaraamistiku `tkinter` õppimisel tahvli koordinaatide mõistmisega ning joonestuselementide paigutamisega tahvlile. Seitsmendal nädalal paaristöö ülesannetes oli tarvis mõista, milline rida on tarvis programmi juurde lisada, et programm teeks nõutud tegevusi.

Üheteistkümnendal nädalal tegelesid paarilised kõige õpitu kordamisega ning neile anti neli programmi, mis olid ridade kaupa lahti lõigatud. Paarilised pidid lahti lõigatud osasid uurides aru saama, mida vastav programm teeb ning osadest kokku panema töötava programmi. Osade paigutamiseks oli tabelkujul alus ette antud. Eelneval nädalal oli õppijatelt küsitud, mis teemal kordavaid ülesandeid nad soovivad.

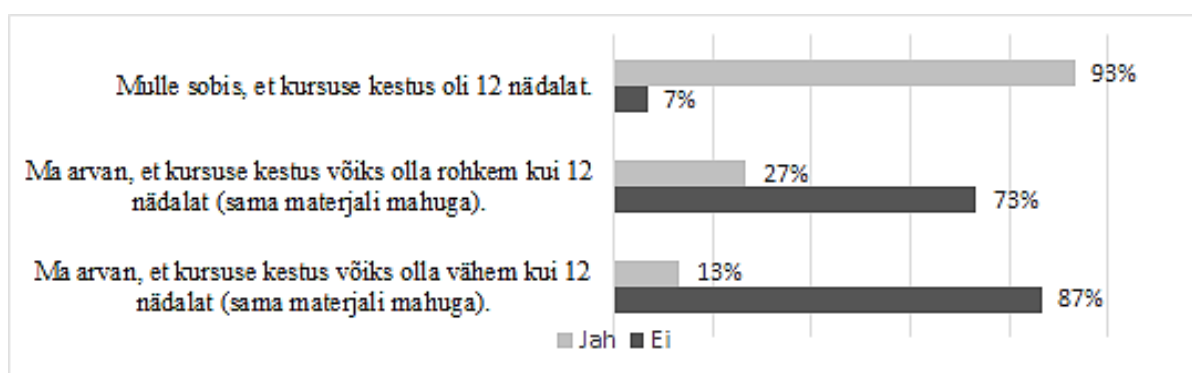
Kõikide tundides paaristööna lahendatavate ülesannete lahendamiseks oli igal nädalal planeeritud ligikaudu 45 minutit, kuid kiiremad jõudsid esimestel nädalatel ka 30 minutiga ülesanded tehtud ning said asuda juba teise tundi planeeritud programmeerimisülesandeid lahendama. Aeglasematel kulus aeg-ajalt ka rohkem aega, kuid sel juhul lubati neil mõni paaristöö ülesanne lahendamata jätta. Valikut tehes jälgis õpetaja, et kõik olulised teadmised saaksid omandatud ning et õppijatel püsiks tahe edasi õppida ja nad ei tunneks end mahajäänutena. Oluline oli jälgida, et kõik jõuaksid tunnis ka programmeerimisülesandeid lahendada, kuna nende lahendamist ei olnud planeeritud koduseks tööks jätta. Tunnist puudujatel tuli seda siiski teha.

Lõpüküsitluses küsiti õppijate hinnangut oma tunniülesannete lahendamisele kulunud ajale ning saadi tulemuseks keskmiselt 69 minutit, mis aga töö autori arvates ei näita tegelikkust. Põhjuseks on ilmselt õppijate küsimuse erinev tõlgendamine ning tunnis tehtud paaristöö ülesannete mitte lugemine kohustulikult lahendatavate ülesannete hulka. Pigem paistab, et õppijad hindasid oma Moodle'i ülesannete lahendamisele kulunud aega, kuna mitte ühelgi nädalal ei saanud keegi õppijatest nii paaristöö kui Moodle'i programmeerimisülesandeid

Väited	Min	Max	Keskmine	Standardhälve
Saan ennast arendada.	3	5	4,53	0,64
Saan tulevikus paremini konkureerida tööturul.	3	5	4,20	0,68
Saan endale esitada väljakutse.	1	5	4,00	1,20
Saan alusteadmised, et programmeerimisega edasi tegeleda.	2	5	3,87	1,06

Küsitluses antud väidetele tuginedes motiveeris õppijaid kursusele tulema kõige enam uute teadmiste saamine (keskmine 4,73; standardhälve 0,46). Huvi programmeerimise teema vastu (keskmine 4,53; standardhälve 0,64), enese arendamine (keskmine 4,53; standardhälve 0,64) ja võimalus tulevikus paremini konkureerida tööturul (keskmine 4,20; standardhälve 0,68), olid õppijate endi hinnangul samuti olulised motivaatorid. Vähem oluliseks pidasid õppijad endi hinnangul endale väljakutse esitamist (keskmine 4,00; standardhälve 1,20) ning alusteadmiste saamist programmeerimisega edasi tegelemiseks (keskmine 3,97; standardhälve 1,06). Seega saab väita, et kõik kursusel osalejad olid programmeerimise õppimisest huvitatud või arvasid, et saadavad teadmised tulevad neile tulevikus kasuks.

Lõpüküsitluses andsid õppijad hinnanguid kursuse ajalise jaotuse sobivuse kohta endale. Kursuse pikkusega, milleks oli 12 nädalat, oli kursuse lõpus rahul 93% õppijatest ja ei olnud rahul 7% õppijatest (joonis 11).



Joonis 11. Õppijate hinnangud kursuse ajalise ülesehituse kohta.

Kommentaarina oli õppijatel võimalik küsitluses oma antud arvamust põhjendada ning selles avaldas üks õppija arvamust, et kursuse pikkus võiks olla sama mahu juures 15 nädalat, kaks õppijat olid küll rahul 12 nädala pikkusega, kuid avaldasid ka arvamust, et kursus võiks olla 14 nädalat või 15 nädalat pikk sama mahu juures. Üks õppija arvas, et kursus võiks kesta

lühemalt, kuid ei täpsustanud oma arvamust. Sellega, et kursus võiks kesta sama mahu juures rohkem kui 12 nädalat, ei nõustunud 73% õppijatest ja nõustus 27% õppijatest ning väitega, et kursus võiks sama mahu juures kesta vähem kui 12 nädalat ei nõustunud 87% õppijatest ja nõustus 13% õppijatest. Seega, kuigi osad õppijad pakkusid ka erinevaid kursuse ajalisi kestusi, oldi siiski kokkuvõttes rahul 12 nädalase kestusega.

Täpsemalt uuriti lõpuküsitluses õppijate hinnanguid auditoorsete tundide ja e-õppe tegevuste sobivuse kohta. Auditoorsete tundide läbiviimisenä paaristunnina olid rahul kõik õppijad, samuti olid kõik rahul auditoorsete tundide ülesehitusega, st esimese tunni ajal paaristöö ja teise tunni ajal iseseivate ülesannete lahendamisega. Seega õppijatele sobis selline lähenemine õpetamisele, kus igal nädalal toimus üks paaristund, millest esimeses kasutati aktiivõppemeetodeid ja tehti MOOC-i materjalidest erinevaid loetud teemade kinnistamise ülesandeid.

Ümberpööratud klassiruumi elemendi, kodus e-õppena sooritatud auditoorseks tunniks ettevalmistava koduse lugemise endale sobivusega pigem nõustuti (keskmine 3,40; standardhälve 1,12), ebasobivaks endale ei märkinud ükski õppija (tabel 5).

Tabel 5. Õppijate hinnangud e-õppe tegevusele.

Väited	Min	Max	Keskmine	Standardhälve
Mulle meeldis, et sain enne tundi ise valida aja materjalide läbitöötamiseks (nt. alustada ühel päeval ja lõpetada teisel).	2	5	4,07	0,96
Ma arvan, et kui tunniks mitte valmistuda, on tunnis raskem teemast aru saada.	1	5	3,53	1,13
Mulle sobis, et uue materjali pidi läbi töötama enne tundi.	2	5	3,40	1,12
Mulle oleks meeldinud, kui ise materjalide lugemise asemel oleks õpetaja tunnis uut teemat seletanud.	1	5	3,20	1,32
Mulle oleks meeldinud, kui koduse lugemise asemel oleks olnud lisatund koolis.	1	4	1,80	1,01

Õppijad nõustusid (keskmine 4,07; standardhälve 0,96) väitega, et talle meeldis võimalus enne tundi ise aeg valida materjalide läbitöötamiseks (nt alustada ühel päeval ja lõpetada teisel). Siiski leidis ka iseseisva lugemise asemel õpetaja seletamist soovijaid (keskmine

3,20; standardhälve 1,32). Täpsustava väitega, kas e-õppe asemel ühe auditoorse tunni lisamise võimalus oleks meeldinud, pigem ei nõustunud (keskmine 1,80; standardhälve 1,01). Seega, kui õppija saab valida koolis lisatunni toimumise või e-õppe osa vahel, siis sellel kursusel osalejad eelistaksid pigem e-õpet. Tunniks valmistumist peetakse oluliseks. Seda näitab nõustumine väitega, et kui tunniks mitte valmistuda, on tunnis raskem teemast aru saada (keskmine 3,53; standardhälve 1,13).

Seega, kõik kursusel õppijad olid pigem rahul kursuse e-õppe osaga ning antud hinnangute põhjal saab väita, et osaliselt e-õpet kasutav kursus sobis nii põhikooli viimases klassis kui gümnaasiumis õppivale õpilasele. Õppijate õpistiilid on erinevad ning mõnele õppijale sobib iseseisev valmistumine rohkem, teisele vähem. Töö autori arvates on iseseisev materjali läbitöötamise meeldivus seotud ka õppijate enesedistsipliini ja -motiveeritusega ning kuna kooli üheks eesmärgiks on õppijates arendada iseseisva töö oskusi ja valmisolekut elukestvaks õppeks, siis ei ole saadud arvamused pigem toetavad ümberpööratud klassiruumi elementide kasutamisest õppetöös.

Õppijate hinnangud e-õppe materjalidele olid positiivsed, vähest huvi tunti üksnes õppevideote vastu (tabel 6).

Tabel 6. Õppijate hinnangud materjalidele ja kodustele tegevustele.

Väited	Min	Max	Keskmine	Standardhälve
Enesekontrolli küsimused materjalide juures olid mulle kasulikud.	2	5	3,71	0,99
Silmaringi materjale oli huvitav lugeda.	1	5	3,17	0,94
Katsetasin läbi kõik materjalides olnud näiteprogrammid.	2	5	3,20	1,15
Programmeerimise materjalid olid tekstiliselt liiga pikad	1	4	2,47	0,92
Vaatasin kõiki õppevideoid.	1	4	1,73	0,88

Õppijatele antud väidetest kõige kõrgema keskmisega oli õppijate hinnang enesekontrolli küsimuste kasulikkuse kohta materjalide materjalides (keskmine 3,71; standardhälve 0,99). Selle väitega mittenõustujaid ei olnud. Väitega, et silmaringi materjale oli huvitav lugeda, oldi pigem nõus (keskmine 3,17; standardhälve 0,94). Väitega, et katsetas läbi kõik materjalides olevad näiteprogrammid, pigem nõustuti (keskmine 3,20; standardhälve 1,15).

Töö autor arvab, et tulemus on seotud sellega, et näiteprogrammidest katsetati läbi vaid osa, mitte kõik. Hinnangut, kui suur osa näiteprogrammidest läbi katsetati, õppijatelt ei küsitud. Programmeerimise materjalide tekste liiga pikaks pigem ei peetud (keskmine 2,47; standardhälve 0,92). Töö autori arvates on tulemus seotud sellega, et MOOC-i kursuse nädala materjalide hulgast tehti õppijatele iseseisvaks lugemiseks valikud osade kaupa lugemiseks ning püüti jälgida, et e-õppe materjali lugemiseks ei kuluks keskmise lugemiskiirusega õppijal rohkem kui 45 minutit. Kõikide õppevideote vaatamisega pigem ei nõustunud (keskmine 1,73; standardhälve 0,88). Madalast keskmisest saab järeldada, et materjalides olevaid videoid kursusel õppijad pigem ei vaadanud või vaadati osas, aga mitte kõiki. Töö autori arvates oli see tulemus oodatav, kuna selgusetu osa kohta küsiti selgitust pigem koolis õpetajalt või arutati oma paarilisega. Koduse lugemise osa võeti kui lihtsalt lugemist, mitte selgeks õppimist ning segaste kohtade selgituste küsimine jäeti auditoorse tunni tegevuseks.

Lisaks andsid õppijad teise nädala tagasiside küsitluses hinnangu oma eelistusele lugeda teksti või slaide. Selle hinnangu järgi eelistas 60% õppijatest slaidide lugemist. Sellele hinnangule tuginedes võiks töö autori arvates kaaluda olemasolevate õppematerjalide täiendamist, kuid ajapuudusel seda selle kursuse raames ei tehtud.

Lisaks e-õppe materjalidele hinnangu andmisele uuriti ka õppijate hinnanguid auditoorsete tundide tegevustele (tabel 7).

Tabel 7. Õppijate hinnangud tunnis tehtud tegevustest.

Väited	Min	Max	Keskmine	Standardhälve
Mulle sobis paaristöö.	3	5	4,62	0,65
Paarilisega koos arutamine aitas mul teemast aru saada.	3	5	4,33	0,82
Paaristöö ülesanded olid mulle kasulikud programmeerimisülesannete lahendamiseks.	3	5	4,13	0,74
Kohustuslikud programmeerimisülesanded olid mulle keerulised.	1	5	2,73	1,03
Kohustuslikke programmeerimisülesandeid oli minu jaoks liiga palju.	1	4	2,40	1,18

Paaristöö sobis kõikidele õppijatele (keskmine 4,62; standardhälve 0,65), selle väitega mitte nõustujaid ei olnud. Samuti saab õppijate hinnangutele tuginedes väita, et paarilisega koos arutamine aitas õppijatel teemast aru saada (keskmine 4,33; standardhälve 0,82) ning

paaristöö ülesanded olid õppija jaoks kasulikud ka programmeerimisülesannete lahendamisel (keskmine 4,13; standardhälve 0,74). Õppijate hinnangul kohustuslikud programmeerimisülesanded pigem ei olnud keerulised (keskmine 2,73; standardhälve 1,03) ning neid ei olnud liiga palju (keskmine 2,40; standardhälve 1,18). Kõik õppijad, kes pidasid ülesandeid keeruliseks, olid teistest aeglasemad ning neil jäi aeg-ajalt mõni ülesanne tegemata, mistõttu kursuse lõpu poole oli nende koormus teistest suurem. Kokkuvõttes saab väita, et aktiivõppe meetodi kasutamine auditoorses tunnis oli õppimist toetav.

Kursuse lõpuküsitluses küsiti õppijate hinnanguid oma tüüpilisele ühe nädala tegevuste järjekorrale. Kui väidetud tegevust üldse ei tehtud, siis oli järjekorra numbriks arv 0, esimesena tehtud tegevus järjekorranumbriga 1 ning viimane järjekorranumbriga 8. Tulemusi analüüsid selgus, et kõik õppijad ei olnud oma tegevusi järjestanud erinevate järjekorranumbritega, näiteks üks õppija oli kõikide tegevuste järjekorranumbriks märkinud arvu 5. Õppijate endi antud tegevuste järjekordade numbreid analüüsid ei muudetud, kuna erisus järjekorra tulemust ei muutnud. Analüüsid leiti antud järjekorranumbrite moodid ja summad, saadud tegevuste järjekord on toodud tabelis 8.

Tabel 8. Õppijate hinnang ühe nädala tegevuste alustamise järjekorrale.

Jrk.nr.	Teemad	Mood	Summa kogusummast
Ei	Videote vaatamine	0	13
1.–2.	Programmeerimismaterjalide lugemine	1	31
	Tunnis tehtud paarisülesannete lahendamine	1	34
3.–5.	Programmeerimise kohustuslike ülesannete lahendamine	3	56
	Murelahendajate kasutamine	5	57
	Mujalt materjali lugemine	0	59
6.	Nädala test	4	72
7.	Silmaringi materjalide lugemine	8	88

Ootuspäraselt loeti esimesena iseseisvalt uue teema materjale, st tegeleti e-õppe tegevustega enne tundi ning seejärel lahendati kooli esimese tunni ajal paaritöö ülesandeid. Oluline on tähele panna, et õppijate e-õppe ja paaristöö tegevuste järjekorrad ei erine suurel määral, mis võib olla autori arvates seotud sellega, et paarisülesannete lahendamise ajal loeti abi

leidmiseks uuesti e-õppe materjale. Küsimustikus ei olnud täpsustatud materjalide lugemist enne tundi e-õppena.

Kolmandana lahendati kohustuslikke programmeerimisülesandeid ning ülesandeid lahendades kasutati vajadusel murelahendajate abi ning loeti ka mujalt materjale. Tulemusi analüüsid saab väita, et need kolm tegevust moodustavad koos tehtud tegevuste grupi. Viimasena lahendati nädala test neil nädalatel, kui see oli olemas ja loeti lisaks „Silmaringi” materjale. Videoid vaatas enda väitel vaid kaks õppijat ning see märgiti ühe vastaja poolt tegevuste järjekorras viimaseks ja teise vastaja tegevused olid märgitud kõik võrdväärsseteks. Seega kokkuvõttes saab väita, et tegevused saab pigem jaotada koos tehtavatesse tegevuste gruppidesse.

Lõpüküsitluses andsid õppijad hinnanguid oma õppimisele kursusel (tabel 9).

Tabel 9. Õppijate hinnangud oma õppimisele kursusel.

Väited	Min	Max	Keskmine	Standardhälve
Küsisin abi, kui jäin ülesannetega hätta.	3	5	4,33	0,72
Kursuse läbimine oli mulle sobivaks väljakutseks.	3	5	4,27	0,59
Ma õppisin sellel kursusel palju.	3	5	4,27	0,70
Teemade valik sobis mulle hästi.	3	5	4,13	0,64
Tundsin õppimisest naudingut.	3	5	4,13	0,74
Süvenesin programmeerimise teemadesse põhjalikult.	2	5	3,80	0,86

Õppijatele antud väidetest kõige kõrgema keskmisega oli õppijate hinnang ülesannetega hätta jäämise korral abi küsimisele (keskmine 4,33; standardhälve 0,72). Selle väitega mitte nõustujaid ei olnud. Samuti peeti kursuse läbimist omale sobivaks väljakutseks (keskmine 4,27; standardhälve 0,59) ning nõustuti väitega, et kursusel õpiti palju (keskmine 4,27; standardhälve 0,70). Teemade valiku sobivusega hinnanguga võib töö autor samuti rahule jääda (keskmine 4,13; standardhälve 0,64). Õppimisest tundi enda hinnangul naudingut (keskmine 4,13; standardhälve 0,74) ning programmeerimise teemadesse süveneti põhjalikult (keskmine 3,80; standardhälve 0,86). Kokkuvõttes saab väita, et õppijate hinnangud oma

õppimisele olid väga positiivsed, mis näitab töö autorile, et auditoorsete tundide läbiviimine oli õppijat toetav ning tehtud teemade valikud sobisid õppijatele.

Lõpuküsitluses andsid õppijad hinnanguid programmeerimisele (tabel 10).

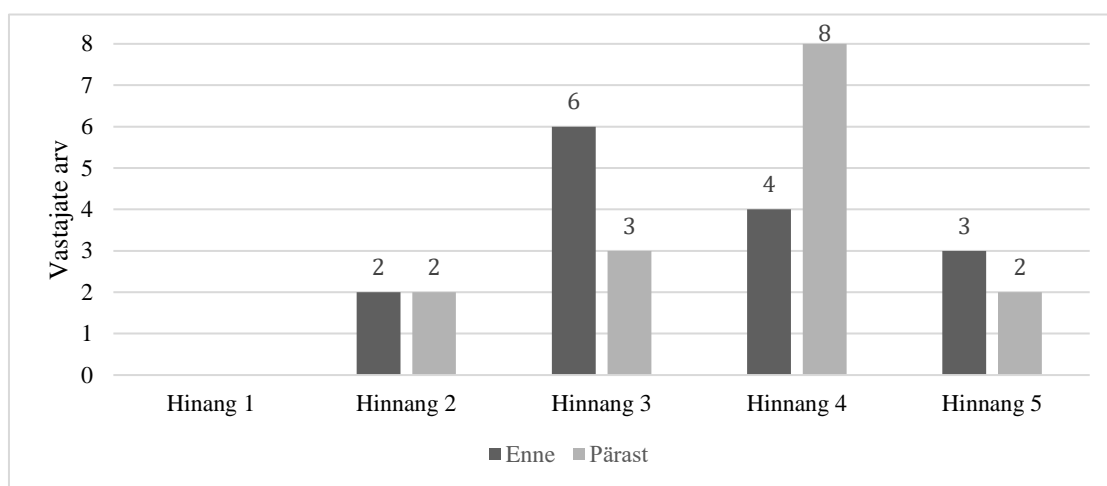
Tabel 10. Õppijate hinnangud programmeerimisele.

Väited	Min	Max	Keskmine	Standardhälve
Ma arvan, et minu jaoks kursus õnnestus.	2	5	4,27	0,88
Ma arvan, et kui sellele kursusele tuleks samalaadne jätkukursus, siis ma osaleksin.	3	5	4,13	0,74
Ma arvan, et sain programmeerimisega hästi hakkama.	2	5	3,73	0,90
Ma arvan, et programmeerimine on mulle sobiv ala.	2	5	3,67	1,11
Ma arvan, et jätkan programmeerimise õppimist ka tulevikus.	1	5	3,67	1,24
Kui peaksin praegu valima edasiõppimise/eluala, siis valiksin suure tõenäosusega programmeerimisega seotud ala.	1	5	3,60	1,42

Õppijatele antud väidetest kõige kõrgema keskmisega oli õppijate hinnang kursuse õnnestumisele enda jaoks (keskmine 4,27; standardhälve 0,88). Samalaadsel jätkukursusel, kui see tehtaks, osalemisele antakse positiivne hinnang (keskmine 4,13; standardhälve 0,74). Programmeerimisega saadi enda hinnangul pigem hästi hakkama (keskmine 3,73; standardhälve 0,90) ja väitega, et programmeerimine on õppijale sobiv ala, pigem nõustuti (keskmine 3,67; standardhälve 1,11). Tulevikus programmeerimise õppimise jätkamise väitega oli nii nõustujaid kui mitte nõustujaid, nõustujaid veidi rohkem kui mitte nõustujaid (keskmine 3,67; standardhälve 1,24). Programmeerimisega seotud edasiõppimise ala valiku väitega oli nii nõustujaid kui mitte nõustujaid (keskmine 3,60; standardhälve 1,42). Programmeerimist oma tulevase erialaga mitteseostav õppija oli kursusel üks edukamatest õppijatest, kes arvas ka, et programmeerimine pole tema ala, kuid hindas siiski oma programmeerimisega hakkama saamist väga heaks. See näitab töö autorile, et kuigi õppija ei plaani programmeerimise valdkonnaga sügavuti edasi tegeleda, pidas ta programmeerimise teadmiste omandamist enda jaoks oluliseks. Kokkuvõttes saab väita, et huvi programmeerimist õppida oli õppijatel olemas ning toimunud kursusega oldi rahul.

Kursuse lõpuküsitluses arvas 71% õppijatest, et kursuse läbimiseks ei ole tarvis eelnevaid programmeerimisalaseid eelteadmisi. Seega nii töö autori kui õppijate arvates sobib antud kursus õpetamiseks põhikooli 9. klassis ja gümnaasiumis ilma eelneva programmeerimisalase kogemusega õppijale.

Õppijate endi hinnangud, kas ta saab programmeerimisega hästi hakkama enne kursuse läbimist ning kas ta sai programmeerimisega hästi hakkama pärast kursuse läbimist, skaalal 1-st (ei nõustu) kuni 5-ni (nõustun) on toodud joonisel 12.



Joonis 12. Õppijate hinnangud oma hakkamasaamisele enne ja pärast kursust.

Kursuse algul arvas kolm õppijat, et saab programmeerimisega väga hästi hakkama, kursuse lõpuks oli selliseid õppijaid kaks, kusjuures need õppijad olid samad, kes kursuse alguseski. Kõrgemalt hindas oma läbimist ootusega võrreldes 40% õppijatest, madalamalt 27% õppijatest ning arvamus ei muutunud 33% õppijatest (joonis 13).



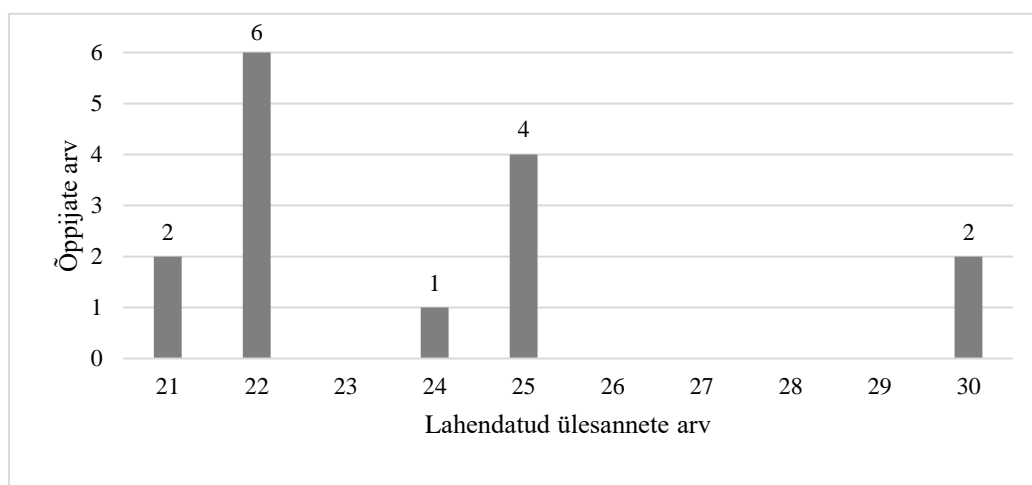
Joonis 13. Õppijate hinnangute muutus enne ja pärast kursust.

Seega õppijate hinnangud oma programmeerimisega hakkama saamisele olid peale kursust pigem natuke paremad, erinevused olid skaalal vaid ühe skaalaväärtuse võrra.

Töö autori jaoks oli huvipakkuvam see, et kursuse viiest edukaimast õppijast vaid kaks hindas oma programmeerimisega hästi hakkama saamist enne ja pärast kursust skaala kõrgeima väärtusega, teised neist ei andnud oma hinnanguks kõrgeimat väärtust ja üks isegi langetas hinnangu väärtust. See võib anda märku sellest, et õppija tundis, et kursus pakkus talle väljakutset ning pani tema teadmised ja oskused proovile.

4.2.2 Ülesannete lahendamise analüüs

Kogu kursuse jooksul oli õppijatel kohustuslik lahendada kokku 21 programmeerimisülesannet (joonis 14). Ainult kohustuslikud 21 ülesannet lahendas kaks õppijat, üks neist oli õppija, kes õpib matemaatikat vähendatud õppekava järgi ning teine neist õppija, kes puudus kokku neljal nädalal, mistõttu jäi teistest maha. Viiel nädalal oli õppijale antud osaliselt võimalus valida lahendamiseks etteantud ülesannete hulgast meeldivam ülesanne, neist kohustuslik oli lahendada vähemalt üks ülesanne (valikülesannetel on ülesande numbril juures tähised a, b või c). Rohkem valiti esimesena antud ülesannet.

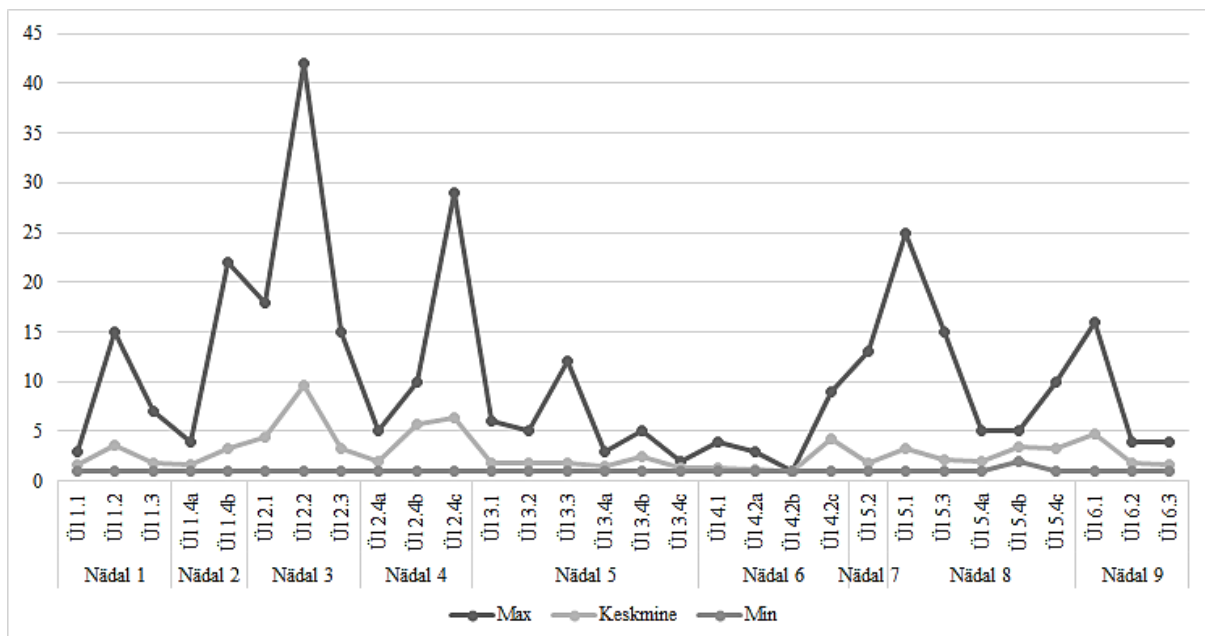


Joonis 14. Esitatud programmeerimisülesannete arvud kogu kursuse jooksul.

Kaks õppijat, kes olid ka ühed edukamatest kursusel, lahendasid kõikide nädalate 1 kuni 9 kõik ülesanded, ehk kokku 30 ülesannet. Lisaks esimese üheksa nädala ülesannetele lahendas veel kolm kiiremat õppijat 10. nädalal kolm lisaülesannet, kuid ühelgi neist õppijatest ei olnud kõik eelnevate nädalate ülesanded lahendatud ning kõik kolm lahendasid kokku 25 ülesannet. Kõik õppijad, kes lahendasid rohkem kui 22 ülesannet, lõpetasid kursuse edukalt hindegas „5”. Õppijatest, kes lahendasid kursuse jooksul 22 ülesannet, ei vajanud üks õppija õpetaja vihjeid kursuse lõputööd lahendades, teised olid ebakindlamad ning vajasisid rohkem

suunamist. Kokkuvõttes võib näha, et edukamad kursusel osalejad lahendasid ka rohkem lisaülesandeid.

Kõikide nädalate ülesannete esituskordade ülevaade on toodud joonisel 15, kus on kasutatud ülesannete tähistamisel lühendeid, mille selgitused on esitatud töö lisana (IV lisa). Esimeste ülesannete lahendamisel valmistas õppijatele peamiselt raskusi täpselt automaatkontrolli nõuetele vastava lahenduse esitamine, mistõttu olid esimestel nädalatel ülesannete esitamiste arvud keskmiselt suuremad kui 5. ja 6. nädalal. Kuna õppijad ei olnud varem kokku puutunud Moodle'i õpikeskkonnaga ega olnud kasutanud Thonnyt, kulus nende keskkondadega tutvumiseks ning nende kasutamise õppimiseks aega. Samuti vajas harjumist erinevate andmetüüpide ning nende lühendite kasutamine programmeerimiskeeles Python. Erandlik on valikülesanne 5.4b, mille minimaalne esituskordade arv on kaks ning mida lahendas ainult kaks õppijat.

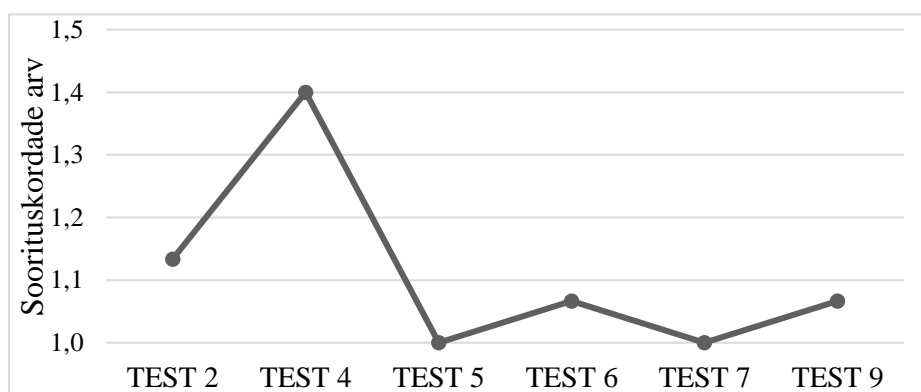


Joonis 15. Ülesannete esituskordade arvud Moodle'i andmetel.

Ühel õppijal oli teise ülesande 2.2 maksimaalne esituskordade arv Moodle'is 42, kui aega kulus sobiva loogilise tehte leidmisele ning Moodle'is esitustele kokku 1,64 tundi. Selle õppija jaoks oli antud ülesanne nõ murdepunkt, edaspidi oli ta juba üks edukamaid ja kiiremaid õppijaid ning tema oli üks neist õppijatest, kes lahendas ja esitas kõik antud programmeerimisülesanded, kokku 30 ülesannet.

Kümnendal nädalal lisaks uusi kohustuslikke programmeerimisülesandeid ei antud, kolm kiiremat õppijat lahendasid lisaülesandeid, kuid teised tegelesid oma pooleli jäänud ülesannetega. Samuti ei olnud 11. nädalal kohustuslikke programmeerimisülesandeid, tunnis tehti kordavaid paaritöö ülesandeid, mille lahendamine võttis mõlema paaristunni aja.

Kokku kuuel nädalal (2. nädal, 4. nädal, 5. nädal, 6. nädal, 7. nädal, 9. nädal) pidid õppijad lahendama testi, mida võis sooritada mitu korda ning mille „arvestatud” hinde saamiseks pidi testi eest saama vähemalt 9 punkti 10-st punktist (joonis 16). Õppijad sooritasid testid hästi, vaid üks kord tegi üks õppija testi kolm korda ning kahel korral said kõik õppijad esimesel testi lahendamise korral kohe testi sooritatud. Kõige paremini sooritati 7. nädala test, kui kõik õppijad said esimesel korral testi lahendades testi eest maksimaalse punktide arvu. Kõige enam kordi, keskmiselt 1,4 korda, tehti 4. nädala testi.



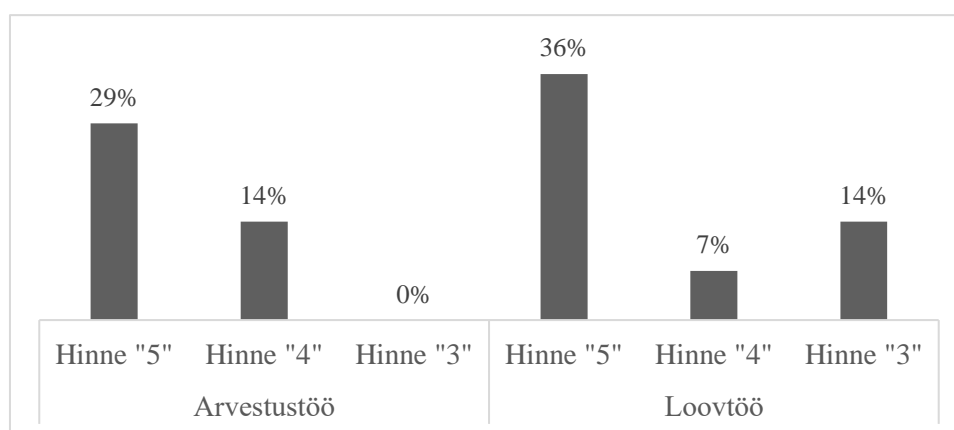
Joonis 16. Testide soorituskordade arvud.

Kaheteistkümnendal nädalal sooritasid õppijad omal valikul kas etteantud arvestusülesande või loovtöö ülesande. Arvestustööks oli MOOC-i kursuse lõpuarvestusülesanne kaltsuvaiba lõimeniidi kogupikkuse arvutamiseks. Arvestustöö lahendamise valis 60% õppijatest, neist 2 õppijat ei saanud ülesande lahenduse eest automaatkontrollilt „arvestatud” tulemust ning neile anti ülesande lahendamiseks vihjeid. Üks õppija puudus koolist lõpuarvestuse tegemise päeval ning sai hiljem teistest erineva arvestusülesande „Käibemaksu arvutamine”.

Arvestustööde puhul kasutati esmakordselt ka sarnasuse analüüsi, mille tulemused näitasid nelja õppija esitatud arvestustööde omavahelist sarnasust. Saadud sarnasuse analüüsi tulemust näidati ka neile neljale õppijale ja küsiti nende arvamust saadud tulemusele, mille peale õppijad tunnistasid, et üks neist lahendas ülesande ära ja teised kirjutasid tema lahenduse enda programmi ümber. Paistis, et sellise sarnasuse analüüsi võimalikkus oli õppijatele, kes kursuse jooksul olid siiski õpetajale positiivselt silma hakanud, ootamatu ning lahendust

otsides lepidi õppijatega nende endi tehtud ettepanekul kokku, et need kolm õppijat, kes iseseisvalt arvestustööd ei lahendanud, teevad lisaks loovtöö.

Seega kokkuvõttes sooritas viimases paaristunnis arvestustöö 6 õppijat ja loovtöö 8 õppijat. Õppija, kes viimasel nädalal alles lahendas oma eelnevate nädalate tegemata ülesandeid ning lõputöö lahendamiseni ei jõudnud, sai võimaluse järele vastamiseks 10 koolipäeva jooksul peale kursuse lõppu. Kolm õppijat vajab oma loovtöö valmimisel õpetaja abi ning hindamise hinnati abivajanud õppijate sooritust hindegaga „3” või „4” sõltuvalt abi suurusest. Kõikide tööd, kes iseseisvalt tegid, hinnati hindegaga „5”. Samuti hinnati hindegaga „5” arvestustöö või lisaülesande eest „arvestatud” saanud õppijaid. Lõputöö valiku ja hinnete ülevaade on toodud joonisel 17.



Joonis 17. Lõputöö valik ja hinnete ülevaade.

Kõikide nädalate ülesanded olid õppijate jaoks saadud teadmiste abil lahendatavad ning raskused ülesannetega või tegemata ülesanded tekkisid neil, kes olid tundidest puudunud ega olnud iseseisvalt teemat endale selgeks teinud. Üks õppija puudus kokku 12-st nädalast 4 korral (1., 6., 9. ja 10. nädalal) ning tema jäi ülesannete esitamisega kõige hilisemaks. Kursuse õpetajale paistis, et kõige kiiremini said teemast aru ning leidsid sobiva lahenduse need õppijad, kes olid varem koolis silma hakanud oma väga heade matemaatika oskustega. Kursuse jooksul 9 nädala tehtud ülesannete eest saadud hinded ning lõputöö hinnete põhjal hinnati kursusel õppijaid kokkuvõtvalt hindegaga „4” (kokku 4 õppijat) või „5” (kokku 10 õppijat). Ühe õppija kokkuvõttev hinne oli „3”.

4.3 Arutelu ja edasised soovitused

Selles alapeatükis arutletakse kursuse õnnestumise üle ning selle üle, mis mõjutab õppija edukust kursuse läbimisel. Samuti antakse soovitusi, kuidas kursust veel paremaks muuta.

Õppimise motivatsioon mõjutab suurel määral ka programmeerimise õppimist. Need õppijad, kes enda väitel olid proovinud ka iseseisvalt programmeerimist õppida, õppisid suurema entusiasmiga ka sellel kursusel ning esitasid õpetajale lisaküsimusi ning ei piirdunud ainult kohustuslike ülesannete lahendamisega. Kursusel anti lahendamiseks ka vabatahtlikke lisaülesandeid, mis algselt olid mõeldud kiirematele, kuid neid lahendasid mõnel juhul ka õppijad, kes olid õppimisest väga huvitatud ja tahtsid end arendada. Töö autori arvates tahtis mõni õppija enesekindluse saamiseks veel harjutamiseks lisaülesandeid. Ühel juhul oli tegemist õppijaga, kes ei plaani programmeerimisega siduda oma tulevikku, kuid oli aru saada, et ta mõistab programmeerimisoskuse olulisust. Seega on programmeerimise algkursust läbi viies väga oluline hoida õppijate õpimotivatsiooni ning anda neile enesekindluse tõstmiseks lahendamiseks vajadusel lisaülesandeid. Oluline on tõsta ülesannete raskusastet sammhaaval, st ülesanded peavad olema õppijate võimetele vastavad, et õppijal püsiks arvamus, et ta saab ülesande lahendamisega hakkama. Lisaks aitab õppijate motivatsiooni kõrgemal hoida rääkimine programmeerimisoskuse väärtusest edasiõppimisel ja tööturul. Oluline on ka leida lahendamiseks selline ülesannete arv, et õppijad ei tunneks, et ülesandeid on liiga palju.

Matemaatika oskamine mõjutab ülesande lahendamise skeemi leidmist, eriti kui ülesandes on tarvis teha mingi tingimuse kontrollimiseks arvutustehteid, näiteks paaris- ja paaritute arvude leidmine. Kursusel kasutatud ülesannete hulgas oli selliseid ülesandeid, kus tuli kasutada matemaatika teadmisi ning selliste ülesannetega said kiiremini ja probleemideta hakkama just need, kelle matemaatilised teadmised on head. Kursusel osalejate hulgas oli ka matemaatikas vähendatud õpimahuga õppiv õppija ning matemaatikas õpiabi saavad õppijad. Neil tekkisid raskused just ülesande teksti mõistmisega ning teksti tõlgendamisega. Tihti vajasisid nad õpetaja abi ja vihjeid lahendusskeemi lahti seletamisel ning pidasid raskemateks just neid ülesandeid, kus oli tarvis teha mõnda matemaatilist tehet, näiteks protsentarvutust. Seega kursusel õppijate edasiminekut vaadates saab väita, et programmeerimise algteadmiste omandamist ei takista vähesed matemaatilised teadmised, kuid sellistele õpilastele tuleb harjutamiseks anda pigem ülesandeid, kus ei ole tarvis kasutada keerulisemaid matemaatilisi tehteid. Seega võib esineda matemaatikaga raskustes olevatel õppijatel raskusi. Autori

tähelepanekul jõudsid programmeerimise kursusel kiiremini ja edukamalt edasi need õppijad, kes on varasemalt silma hakanud oma väga heade teadmistega matemaatikas.

Varasem vähene kokkupuude programmeerimisega annab eelise esimestel tundidel. Kui esmased teadmised on omandatud, siis erinevus varem programmeerimisega kokku puutunute vahel väheneb. Läbiviidud kursuse õppijate hulgas ei olnud keegi varasemalt programmeerimiskeelt Python õppinud, seega ei olnud kellelgi otsest eelist. Need õppijad, kes olid varem Codeweeki ürituse raames samas koolis tegelenud kilpkonnagraafikaga, said kiiremini aru ka vastavast teemast kursusel. Samuti arvasid kursuse õppijad, et kursuse läbimiseks ei ole varasemad teadmised programmeerimisest vajalikud. Seega kursus sobib päris algajale, kuid sel juhul tuleb sellega arvestada esimestes tundides ning alustada uue info edasiandmist aeglasema tempoga.

Alguses kulub aega ka programmeerimiskeskonna Thonnyga harjumisega ning saadud veateadete mõistmisega. Kuna antud kursuse õppijatel puudus varasem kokkupuude automaatkontrolliga, siis ülesande lahenduse koostamisel nii, et see vastaks ka automaatkontrolli poolt esitatud nõuetele, esines raskusi. Üks olulisemaid esimeste tundide omandatud teadmisi on õppija arusaamine, et programmi kirjutades ei pea kohe veateadeteta tulemus tulema. Õppijad, kes on pigem harjunud teistes ainetes kohe õiget vastust esitama, võivad esialgu olla häiritud, kui programm annab kompileerimisel veateate. Oma vigade leidmine on alguses vaevaline ning mõnikord võis viga seisneda vaid õigekirjas. Selles ongi põhikooli ja gümnaasiumi õpilaste jaoks oluline õpetaja füüsiline kohalolu, kes saab õpetada ka seda, kuidas kontrollida kirjutatud programmi vastavust nõuetele. Algõppe programmid ei ole üldjuhul sisuliselt keerulised ning esimestes programmides tekkivad vead on seotud erinevate andmetüüpide õige kasutamisega, taanetega ning kooloni puudumisega tingimuslauses. Need õppijad, kes on juba veidi programmeerimisega tegelenud, oskavad ise kergemalt selliseid vigu leida ja märgata.

Tundidest puudumine mõjutab koolis suurel määral hakkama saamist, kui õppija ei ole ise valmis kohe oma puudumisest tekkinud teadmiste puudujääke likvideerima. Tihti on puudumine seotud haigestumisega ning sel juhul tekib õpilasel puudujääke ka kõikides teistes õppeainetes ning tervenenult on õpilasel suur koormus teistele järgi jõudmisel. Kuna programmeerimise algkursus oli valikkursus, siis oli töö autorile selge, et põhiainetega tegeles õppija esimeses järjekorras ning programmeerimise kursuse tegemata ülesandeid ei tehtud esimesena. Kõige suuremad probleemid tekkisid kursusel õppijal, kes puudus kokku

neljal nädalal, mis tähendas, et tal jäid õppimisel puudu paaristöö ülesannete arutelud ning oli näha, et see mõjutas tema arusaamist teemadest. Paaril korral oli ühel õpilasel tegemist ka põhjusega puudumistega tundidest ning oli näha, et tema õppimise motivatsioon oli langenud. Töö autor nägi motivatsiooni languses seost ka sellega, et samas klassi käivatel sõpradel edenes õppimine kiiremini ja kergemalt. Kursuse õpetaja ei saa mõjutada haigestumisega tekkivaid puudumisi, kuid vaja on leida võimalus õppija aitamiseks, et tal ei kuhjuks tegemata ülesandeid. Üheks võimaluseks on pakkuda õppijale individuaalset konsultatsiooni, kuid see eeldab ka õppijapoolset tahet. Selleks, et peale puudumist oleks õppijal võimalik järgnevate tundide tegevustega edukalt edasi minna, on tarvis puuduv osa võimalikult kiiresti selgeks teha.

Töö autori jaoks oli positiivne, et kõik õppijad arvasid, et paarisülesannete lahendamisel oli neil kasu ka kohustuslike programmeerimisülesannete lahendamisel. Paaristöö ülesandeid koostades jälgiti teadlikult, et vastavad ülesanded kontrolliksid loetud teemade kõige olulisemaid nüansse. Kõikidele õpilastele meeldis paaristööd teha, kaaslasega arutamine aitas paremini õpitavat mõista ning õppijad läksid aja möödudes aina julgemaks ja enesekindlamaks. Kui paarides olevad õppijad olid teineteisele võõrad, st nad ei olnud ühest klassist, siis algul olid need paarid pigem tagasihoidlikumad ning õppijad ei julgenud nii avatult oma lahendust pakkuda. Edaspidi, kui oldi juba rohkem koostööd tehtud, oldi omavahelistes aruteludes ka julgemad ega kardetud eksida. Lisaks andis võimalus õppematerjalidest vajadusel abi ja lisainfot otsida õppijatele vabaduse ise oma õppimist juhtida ning mitte peljata mingi fakti mittemäletamist. Seega aktiivõppemeetodite kasutamine programmeerimise algõppe kursuse läbiviimisel aitas kaasa õpitava mõistmisele.

Iseseisva e-õppe ehk kodutöö eesmärk on muuhulgas seotud oskusega teadvustada ja suunata iseenda õppimisprotsessi. Lisaks aitab see ühtlustada õppijad taset, st ka aeglasemad õppijad ei jää kiirematest maha ning saavad endale sobivas tempos edasi liikuda. Keegi kursusel osalejatest ei kurtnud koduse e-õppe liiga suure mahu üle. Õppijal oli vabadus ise oma aega planeerida ning valida endale sobiv aeg nädala jooksul, millal materjal läbi töötada. Samas oli oodatav tulemus, et õppija eelistaks ise lugemise asemel õpetaja seletamist, aga mitte selle jaoks lisatundi. Põhikooli ja gümnaasiumi õppijate jaoks on oluline ka koolipäeva pikkus ning nagu selgus vestlustest kursusel õppijatega, on oluline ka see, kas klassikaaslased saavad juba koju minna, aga tema peab veel koolis olema. Samuti võib mõjutada kursusel osalemist teistes huvikoolides osalemine ning ka bussiliikluse graafik. Kursusel õppijatega oli

kokkulepe, et need õppijad, kellel oli tarvis enne auditoorse tunni lõppemist juba lahkuda, võivad seda teha, kui nad iseseisvalt teevad vajalikud ülesanded ära. Probleeme kursuse jooksul sellega ei tulnud, kuid õpetajal on vaja seetõttu olla tähelepanelik selliste õpilaste võimalike õpiprobleemide märkamisel. Vajadusel saab lisaks veebi kaudu konsultatsioone pakkuda või õppijale saadetavate sõnumite kaudu ka probleemsetele kohtadele vihjata.

Kuna kõik kursusel õppijad jäid rahule nii kursuse e-õppe osaga kui ka kogu kursuse korraldusega, siis saab väita, et osaliselt e-õpet kasutav kursus ehk kombineeritud õppe kasutamine sobib kooliõpilasele. Õppijates iseseisva töö oskuste arendamiseks ja ennastjuhtivaks õppimiseks sobib kasutada ümberpööratud klassiruumi elemente. Olemasoleva MOOC-i kursuse materjalid on samuti sobivad kursuse läbiviimisel koolis, vajalik on vaid õppijate kontakt kursuse läbiviijaga, kes suunaks õppijaid edasi liikuma neile sobivas tempos. Töö autori eelnev kogemus ning uuritava kursusega saadud kogemus näitab, et kooliõpilasel on kursuse läbimisel vajalik auditoorsete tundide olemasolu. E-õppe osakaal võiks jääda ligikaudu sama suureks, kui oli uuritaval kursusel, st ligikaudu 30% tundidest võib olla e-õppena. Nii soovitatakse ka erinevates uuringutes. Kas ja kuidas kontrollida, et õppija on e-õppena antud osa õppinud? Läbiviidud kursusele seda ei kontrollitud, kuid töö autor soovitaks edaspidi iga e-õppe osa kohta lasta esitada õppijatel loetu kohta lühikese tagasiside või paluda koostada loetu kontrollimiseks paar küsimust, mida saab ka paaristöö osana kasutada.

Kursust läbi viies on vaja otsustada, kas hindamisel kasutada eristavat või mitteeristavat hindamist. Eristava hindamisskaala kasutamise oluline eesmärk on anda õppijatele täpsemat tagasisidet. Seoses väljundipõhise õppekava lähtumisega õpitulemustest ning õppimist toetava hindamise rolli kasvuga on koolides suurenemas ka mitteeristava hindamise osakaal, kuid programmeerimise algkursuse puhul ei ole töö autori arvates vajalik ega võimalik määrata ülesannete lahendamisel kasutatavate oskuste omandamise piiri. Algteadmised on kõik olulised ning üheks võimaluseks olekski nõuda õppijatelt, et nii nagu MOOC-i kursusel, peab ka kursuse iga õppija saama kõik kohustuslikud ülesanded tehtud ning nädala kokkuvõte peab olema „arvestatud”. Töö autori arvates on algõppe puhul olulisem see, et õppijad omandaksid kõik vajalikud kursusel õpitavad põhiteadmised ning seetõttu teavitati ka kursuse alguses õppijaid hindamise nõudest, et kõik kohustuslikud ülesanded peavad olema sooritatud.

Kuna läbiviidud kursus oli töö autori otsusel eristava hindamisega, siis kaasnes sellega vajadus otsustada, mis teha õppijaga, kellel jääb mõni ülesanne tegemata või jäävad ülesanded vea sisse. Kursuse läbiviimisel jälgiti, et kõikidel õppijatel saaksid kõik kohustuslikud ülesanded tehtud ning Moodle'i keskkonnas oleks iga ülesanne „arvestatud”. Kui kursuse vastava nädal oli kokkuvõttes „arvestatud”, siis hinnati see hindegaga „5”, sõltumata sellest, kas õppija vajab mõne ülesande lahendamisel õpetaja vihjeid ning nõuti, et iga nädal peab saama kokkuvõtvaks hindeks „arvestatud”. Teiseks võimaluseks oleks olnud arvestada iga nädala ülesannete puhul, kas tehtud on vähemalt pooled ülesanded, mis võiks tähendada hinnet „3” või tehtud on vähemalt 75% ülesannetest, mis võiks tähendada hinnet „4”. Sellise hindamisega kaasneksid omad probleemid, näiteks jääks õppijal mõned olulised teadmised ja oskused saamata ning see omakorda võib mõjutada edasist õppimist. Kuna koolis kehtib nõue, et 10 koolipäeva jooksul on õpetajaga kokkuleppel õppijal võimalik tegemata tööd vastata, siis kursuse viimaseks tunniks oli veel ühel õppijal üks ülesanne tegemata ning ühel õppijal veel tegemata lõputöö. Kasutatud hindamismeetod sobis kursusel osalejatele ning kokkuvõtvalt hinnati ühet õpilast hindegaga „3”, nelja õpilast hindegaga „4” ja kümnet õpilast hindegaga „5”. Seega on sarnast kursust läbi viies veel teisi võimalusi hindamise nõuete kokkulepeteks õppijatega.

Hindamisega seotud on küsimus, kuidas lahendada olukorrad, kui õppijad kasutavad ebaausaid võtteid. Hindamise kord on igal koolil sätestatud kooli kodukorras. Põhikooli riiklik õppekava § 21 lõige 6 sätestab, et „kui hindamisel tuvastatakse kõrvalise abi kasutamine või mahakirjutamine, võib kirjalikku või praktilist tööd, suulist vastust (esitust), praktilist tegevust või selle tulemust hinnata hindegaga „nõrk”, kui see on ette nähtud kooli õppekavas.” (Põhikooli riiklik õppekava, 2011). Lõige 7 sätestab, et „kui kirjalikku või praktilist tööd, suulist vastust (esitust), praktilist tegevust või selle tulemust on hinnatud hindegaga „puudulik” või „nõrk” või on hinne jäänud panemata, antakse õpilasele võimalus järelevastamiseks või järeltöö sooritamiseks. Järelevastamise ja järeltööde sooritamise kord sätestatakse kooli õppekavas.” (Põhikooli riiklik õppekava, 2011). Sarnased sätted on ka Gümnaasiumi riiklikus õppekavas § 17 lõige 4 ja lõige 5 (Gümnaasiumi riiklik õppekava, 2011). Kursuse arvestustöös kasutasid neli õppijat ebaausaid võtteid ning olukord lahendati vastavalt kooli kodukorras seatud korrale. Kuna hindamise eesmärgiks on muuhulgas innustada õpilast sihikindlalt õppima ning suunata õpilase enesehinnangu kujunemist, otsustati seekord olukord lahendada leebelt, lootes, et õppijad tegid juhtunust õiged järeldused.

Töö autori jaoks oli kursus kokkuvõttes edukas. Kõik õppijad olid tunnis aktiivsed õppijad, nõrgem leidis paaristööd tehes toe tugevamalt ning kõik õppijad lõpetasid kursuse positiivse hindega. Kursust läbi viies oli töö autorile kõige suuremaks toeks juba olemasolevad õppematerjalid ning Moodle'i kursuse kasutamise võimalus, kuna seal said õppijad kasutada on programmeerimisülesannete kontrollimiseks automaatkontrolle. See andis võimaluse õppijal saada kiiret tagasisidet oma lahendusele, sest ta ei pidanud ootama, kuni õpetaja tema lahendust kontrollida jõuab. Kiire tagasiside õpetas ka õppijaid ise oma lahendustest vigu otsima ja parandusi tegema.

Õppijate motivatsiooni hoidmiseks oli oluline paaristöö ülesannetena anda ka kergemaid ülesandeid. Samuti õnnestus ülesannete kombineerimine eelnevate nädalate teemadega ning seotus programmeerimisülesannetega, et varem õpitu ei ununeks ning ees ootavate ülesannete jaoks oleks neist abi. Töö autori jaoks oli oluline ka õppijate rahulolu teemade valikuga. See näitab, et õppijad tundsid, et teemad on neile jõukohased ning samas ka mitte liiga lihtsad ja pakkusid õppimisel väljakutset. Õppija jaoks on oluline tunda, et ta on iga tund midagi uut omandanud. Õppijate motivatsiooni hoidmiseks jälgiti ka, et õppija ühe nädala koduse töö maht ei oleks liiga suur ning seetõttu ei antud lisaks planeeritud e-õppe osale iseseisvaks lahendamiseks kohustuslikke programmeerimisülesandeid. Siiski oli osadel õppijatel tarvis tunnis pooleli jäänud ülesandeid kodus lõpetada, et mitte teistest maha jääda. Vabatahtlikult lahendatavate programmeerimisülesannete võimalust võiks edaspidi ikkagi kaaluda, kuna ka läbiviidud kursusel oli õppijaid, kes lahendasid lisaülesandeid.

Töö autor arvas enne toimunud kursust, et 12 nädalat järjest iga nädal kolm akadeemilist tundi programmeerimise õppimist võib olla õppija jaoks koormav ning seetõttu õnnestus aja planeerimine selliselt, et koolivaheaeg jäi kursuse keskpaika. Samuti näitab töö autori kogemus, et maikuus, kui ilmad on juba ilusad ja õppijad on kooliaastast väsinud, ei ole õppimine enam nii tulemuslik. Mistõttu oli hea planeerida kursus nii, et see lõppes enne maikuud.

Kõige raskem oli töö autori jaoks leida sobivaid lisavõimalusi tunnist puudujate järele aitamiseks, kui õppija ise abi otsima ei tulnud. Samuti oli tarvis alates kolmandast nädalast pikendada ülesannete esitamise tähtaegu, mis esialgu ei olnud nii planeeritud. Siiski erinevatel põhjustel ei jõudnud kõik õppijad auditoorses tunnis ülesandeid valmis ning iseseisvalt lõpetades tekkivatele probleemidele ei leitud ise lahendust. Kuna aga kursuse eesmärk oli, et iga õppija leiaks lahenduse esitatud ülesannetele, siis alates viiendast nädalast

ei seatud üldse ülesannete esitamisele Moodle'is tähtaega. Töö autori arvates ei olnud see küll kõige parem lahendus, kuid aitas kaasa sellele nõudele, et kursuse lõpuks peavad igal õppijal kõik kohustuslikud ülesanded lahendatud olema.

Samuti jäi töö autori arvates väheseks andekamale ja kiiremale õppijale temale sobiva väljakutse andmine. Osaliselt oli see seotud ka sellega, et töö autor ei oska veel ise automaatkontrolle ülesannetele koostada ning sai kasutada vaid olemasolevaid automaatkontrollidega ülesandeid. Lisaks tekkis kursuse jooksul töö autoril vajadus otsustada, kas anda kiirematele lisaülesandeid samal teemal või lasta neid kursusele mittemahtuvat teemat uurida. Kuna aga hetkel ei ole töö autoril võimalust koolis jätkukursust korraldada, lisaks lõpetavad osad kursusel osalejad põhikooli ning ei plaani jätkata õpinguid gümnaasiumis, siis küsiti ka õppijate endi arvamust ning õppijate soovile tuginedes anti kiirematele lisaülesandeid uuest teemast.

Edaspidi programmeerimise algõpet läbi viies tasuks teha olemasoleva materjali põhjal kaks kursust. Neist esimeses osas mitte käsitleda funktsioonide teemat, sest funktsioonide kasutamise oskus on oluline, kuid see oskus jäi praegusel kursusel väheseks. Funktsioonide teema võiks jääda kursuse teise osasse, kus saaks seda teemat põhjalikumalt käsitleda. Töö autor jaotaks lisaks teema „Sõned” ja „Graafika” kahele erinevale nädalale. Esimeses osas ei oleks tarvis ka teha lõpus arvestustööd. Pigem tasuks eelistada pikema aja jooksul õpilase loovtöö koostamist ja ka teistele esitlemist. Loovtöö võib olla tehtud ka paaristööna, see õpetab õppijaid lisaks oma tööd planeerima ja esitlema ning annab praktilise väljundi kursuselt saadud teadmistele. Kui planeerida kaks kursust, siis võiks need olla samal õppeaastal, esimene osa septembrist detsembrini ja teine osa jaanuarist aprillini.

Programmeerimise algkursuse õpetamine koolis kombineeritud õppena võimaldab koolil arendada õppijate programmeerimise alaseid teadmisi ka juhul kui koolil ei ole programmeerimise õpetajat. Kui osa tunde on e-õppena, siis väljaspool oma kooli õpetajaskonda tulev õpetaja saab kursust õpetada lühema aja vältel intensiivsema õppena või samaaegselt mitmes koolis. Võimalusi on mitmeid ja endale sobivad tuleb igal koolil ja õpetajal ise leida. Kuna läbiviidud kursusel osalejate arv oli 15, mis seab üldistuste tegemisele piirangud, kuid kursuse tulemused näitavad, et sellise kursuse õpetamine saab olla edukas.

5. Kokkuvõte

Selle töö eesmärk oli koostada õppematerjalid Pythoni programmeerimise algõppe kombineeritud õppena läbiviimisel kooli kolmandas kooliastmes või gümnaasiumiastmes valikkursusena kasutades Tartu Ülikooli e-õppe kursuse „Programmeerimise alused” materjale. Samuti kirjeldada koostatud õppematerjalide kasutamise võimalust kursuse läbiviimisel.

Töö antakse ülevaade programmeerimiskeele Python õpetamise võimalustest ja olemasolevatest eestikeelsetest õppematerjalidest. Kuna uuringud näitavad, et Eesti koolidel on ebaühtlane infotehnoloogia alaste oskuste õpetamise tase ning paljude üliõpilaste väljalangevus kõrgkoolidest on seotud üliõpilaste jaoks raskete programmeerimise algkursustega, mistõttu soovitatakse varakult alustada kvaliteetse infotehnoloogia alase õppega koolides. Samuti näitavad uuringud, et koolid vajavad kõrgkoolide abi programmeerimise alaste õppematerjalide koostamisel, mis ajendaski töö autorit uurima võimalust, kuidas kasutada juba olemasolevaid e-õppe materjale koolis programmeerimise kursuse läbiviimisel. MOOC-ide kombineerimine tavakursustega, mida on ülikoolis juba tehtud, on uudne kogu haridussüsteemis ning töö autor soovis näidata, et seda saab edukalt teha ka koolis kasutades kombineeritud õpet ning ümberpööratud klassiruumi elemente.

Enne kursuse algust oli esimeseks tähtsaks otsuseks kursuse ajalise pikkuse valik. Töö autori arvas, et arvestades enamuse kursuse õppijate õppekoormust seoses ees ootavate põhikooli lõpueksamitega, on kõige sobivam lõpetada kursus enne kevadist koolivaheaega, mistõttu kujunes kursuse pikkuseks 12 nädalat. Uuringus selgus, et tehtud valik sobis õppijatele ning 12 nädala pikkuse kursuse pikkusega oldi rahul. Teiseks oluliseks otsuseks oli kursuse e-õppe ja auditoorse õppe jaotus. Kuna koolis on õppekava tihe, ei soovitud õppijate iseseisva töö koormust palju suurendada, mistõttu tehti valik e-õppe osaks jätta ligikaudu kolmandik kursuse tundidest. Töös selgub, et tehtud valik sobis ka õppijatele, kuigi leidis ka neid õppijaid, kes eelistaksid iseseisva õppimise asemel õpetaja selgitusi. Tulemus oli oodatav, kuna töö autorile teadaolevalt ei olnud õppijad varem koolis kombineeritud õpet kasutanud.

Kuna uuringus kasutatud MOOC-i õppematerjalide läbimiseks arvestuslik aeg oli mõnevõrra mahukam kui on koolis ühe kursuse pikkus, siis arvestati kursust planeerides võimalusega, et jooksvalt võib tekkida vajadus teemade läbimise tempot muuta. Esialgu oli siiski plaanis läbida kõik teemad ning jätta vaid lisaõppemise teemad kohustuslike materjalide hulgast välja.

Kursuse jooksul selgus, et kõikide teemade läbimise korral võivad jääda õppijate teadmised pealiskaudseks, mistõttu otsustati töö autori poolt pigem vähendada läbitavate teemade hulka. Seega kursuse aluseks oleva MOOC-i materjalidest ei jõutud läbida MOOC-i seitsmenda nädala teemat „Andmevahetus. Lihtne kasutajaliides”, teised teemad läbiti. Siiski teeb töö autor ettepaneku edaspidi teema „Sõned. Graafika” läbida ühe nädala asemel kahe nädalaga.

Töö autor oli seisukohal, et MOOC-i materjalide läbimisest üksi ei piisa, et kursus oleks õppijate jaoks edukas ning kursuse jooksul õppijate õpimotivatsioon ei kaoks. See ajendas kasutama kursusel aktiivõppe meetodeid ning lisaks e-õppe materjalidele koostama õppijaid abistavaid ülesandeid. Kursuse lõpuks saab väita, et selline lähenemine sobis õppijatele ning töö autor soovib sarnaselt koostatud kursust koolis kasutada. Oluline on leida lahendamiseks selline ülesannete arv, et õppija ei tunneks, et ülesandeid on liiga palju. Läbiviidud kursusel ei kontrollitud, kas õppija on e-õppe ülesanded sooritanud, kuid töö autor soovib uuringu jätkajatel iga e-õppe osa kohta lasta esitada õppijatel tagasiside või leida muu võimalus kontrollimiseks.

Töö autor leiab, et uuringut korrates tasub läbi mõelda uuesti hindamise süsteem. Kursusel ei õnnestunud see, et kõik õppijad esitaksid etteantud tähtjaks nõutud ülesanded, kuid oleks hea, et ükski õppija ei jääks maha ning edasi liikudes ei tekiks probleeme. Kõige raskem oli töö autori jaoks leida sobivaid lisavõimalusi tunnist puudujate järele aitamiseks, kui õppija ise abi otsima ei tulnud. Samuti peab valmis olema kiirematele ja edukamatele väljakutset pakkuma kas lisaülesannete või lisateemadega. Eristavat hindamist kasutades saab luua sobiva hindamissüsteemi, samas mitteeristavat hindamist kasutades sobib MOOC-i Moodle'i keskkonna hindamise arvestus paremini. Hindamisega on seotud ka küsimus, kuidas lahendada olukorrad, kui õppijad kasutavad ebaausaid võtteid.

Seega kursus sobib päris algajale, kuid sel juhul tuleb sellega arvestada esimestes tundides ning alustada uue info edasiandmist aeglasema tempoga. Huvitav oleks uurida, kuidas sobib kursus neile, kes on varasemalt programmeerimist õppinud, näiteks kooli huviringis. Kas nende jaoks, kes on näiteks Scratchiga programmeerimist õppinud, on ka kursus sobiva teemade valikuga? Töö autori jaoks oli huvitav märgata, et need õppijad, kes on varasemalt silma paistnud oma väga heade teadmistega matemaatikas, on edukamad ka programmeerimises.

Järgnevate sarnaste uuringute planeerimisel ja läbiviimisel soovitab autor teha MOOC-i olemasoleva materjali põhjal kaks kursust. Neist teise osasse viia funktsioonide teema, kuna funktsioonide kasutamise oskus on oluline, kuid see oskus jäi töö autori hinnangul praegusel kursusel väheseks. Töö autor jaotaks lisaks teema „Sõned” ja „Graafika” kahele erinevale nädalale. Kursuse esimese osas ei ole vajadust teha lõpus arvestustööd. Pigem tasuks eelistada pikema aja jooksul õpilase loovtöö või rühmatöö koostamist ja ka teistele esitlemist. Kui planeerida kaks kursust, siis võiks need olla samal õppeaastal, esimene osa septembrist detsembrini ja teine osa jaanuarist aprillini.

Lõpuküsitluse tulemuste analüüsimisel segus, et kui tahta teada õppijate ühe nädalal tegevuste sooritamise järjekorda, siis läbiviidud küsitluse küsimust tuleb muuta. Ilmnes, et osad õppijad ei järjestanud oma kõiki tehtud tegevusi järjest, Üks õppija märkis kõikide tegevuste järjekorraks skaala väärtuse 5 ning oli ka õppijaid, kes jätsid mõne skaala väärtuse vahele.

Töö autor täiendas Courses-keskkonna lehel esimese nädala esimese teema materjale slaidiesitlusega, kuid ajapuudusel teiste nädalate materjalide hulgas muutusi ei tehtud. Siiski esimese nädala tagasisides märkisid ligikaudu pooled õppijad, et neile meeldiks slaide rohkem lugeda kui teksti. Samuti ei muudetud sealse keskkonna nädalate jaotust, kuid sama uuringut uuesti läbi viies, tasuks mõelda, kas seda siiski teha.

Tehtud töö pakub ühe lahenduse, kuidas programmeerimise algõppe kursust koolis läbi viia. Kuna uuringud näitavad, et koolides on programmeerimise õpetajate puudus, siis kombineeritud õppe kasutamine pakub koolile ühe lisavõimaluse õpilaste programmeerimisoskuste arendamisel. Kui osa tunde on e-õppena, siis väljaspool oma kooli õpetajaskonda tulev õpetaja saab kursust õpetada lühema aja vältel intensiivsema õppena või samaaegselt mitmes koolis. Võimalusi on mitmeid ja endale sobivad tuleb igal koolil ja õpetajal ise leida.

Kuna läbiviidud kursusel osalejate arv oli 15, siis see seab üldistuste tegemisele piirangud, kuid kursuse tulemused näitavad, et sellise kursuse kasutamine sobib koolis. Selgelt eristuvaid erinevusi 9. klassi ja 10. klassi õpilastele sobivuses ja kursuse edukuses ei ilmnenu. Läbiviidud kursus oli kursusel õppijate ning töö autori arvates edukas ning kursuse läbiviimine pakkus autorile väljakutset oma erialaste teadmiste parandamiseks ning

kompetentsuse arendamiseks uue õpimudeli rakendamisel. Lisaks teadmiste andmisega, arendas kursusel osalemine õppijate ajajuhtimis-, meeskonnatöö ning analüüsioskusi.

Töö autor on tänulik Tartu Ülikoolile, kes andis võimaluse kursuse läbiviimisel kasutada Tartu Ülikooli Moodle'i õppekeskkonda ja Courses-keskkonda. Moodle'i hindamissüsteemi ümbertegemine läbiviidava kursuse jaoks pakkus väljakutset ka autorile. Kuna olemasolev MOOC-i kursuse jaoks on välja töötatud programmeerimisülesannete automaatkontrollid, siis nende kasutamise võimalus lihtsustas olulisel määral kursuse läbiviimist. Autor loodab, et ka uuringu jätkajad saavad sama automaatkontrollide võimalust kasutada ning tulevikus on kõikidel koolidel võimalik sarnast automaatkontrollide võimalust kasutada.

6. Kasutatud kirjandus

Annamaa, A. (s.a.) Tartu Ülikooli Arvutiteaduse instituudi programmeerimise algkursuse õpik. <http://progeopik.cs.ut.ee> (14.03.2018)

Annamaa, A. (2015). Thonny: a Python IDE for Learning Programming. *In Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, p. 343–343. <https://dl.acm.org/citation.cfm?id=2754849> (07.03.2018)

Aramaa, A. (2014). Pythoni koodi muudatuste analüsaator. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö.

Ateeq, M., Habib, H., Umer, A., Rehman, M. U. (2014). C++ or Python? Which One to Begin With: A Learners Perspective. *2014 International Conference on Teaching and Learning in Computing and Engineering*, p. 64–69.

Baker, J. W. (2000). “The “classroom flip“: Using web course management tools to become the guide by the side”. *Communication Faculty Publications*. 15. http://digitalcommons.cedarville.edu/media_and_applied_communications_publications/15 (14.03.2018)

Barg, M., Fekete, A., Greening, T., Hollands, O., Kay, J., Kingston, J. H. (2000). “Problem-Based Learning for Foundation Computer Science Courses”. *Computer Science Education*, vol. 10(2), p. 109–128.

Beaver, J. K., Hallar, B., Westmaas, L. (2014). Blended Learning: Defining Models and Examining Conditions to Support Implementation. *PERC Research Brief*. <http://www.researchforaction.org/wp-content/uploads/2015/11/Blended-Learning-PERC-Research-Brief-September-2014.pdf> (14.03.2018)

Bishop, J. L., Verleger, M. A. (2013). The Flipped Classroom: A Survey of the Research. *120th American Society of Engineering Education Annual Conference & Exposition*. http://sthembrazil.com/site/wp-content/uploads/2016/05/The_Flipped_Classroom_A_Survey_of_the_Research.pdf (14.03.2018)

- Boelens, R., De Wever, B., Voet, M. (2017). Four key challenges to the design of blended learning: A systematic literature review. *Educational Research Review*, 22, p. 1–18. <https://biblio.ugent.be/publication/8526490/file/8526491> (12.03.2018)
- Bonk, C. J., Graham, C. R. (2006). The handbook of blended learning: Global perspectives, local designs. United States of America: John Wiley & Sons.
- Bruff, D. O., Fisher, D. H., McEwen, K. E., Smith, B. E. (2013). Wrapping a MOOC: Student Perceptions of an Experiment in Blended Learning. *MERLOT Journal of Online Learning and Teaching*. June, Vol. 9, No. 2. http://jolt.merlot.org/vol9no2/bruff_0613.htm (14.03.2018)
- Cheang, K. I. (2009). Effect of Learner-Centered Teaching on Motivation and Learning Strategies in a Third-Year Pharmacotherapy Course. *American Association of Colleges of Pharmacy*, 73(3):42. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2703280/> (07.03.2018)
- Dixon, N. (2013). Scaffolding Fully Online First Year Computer Literacy Students for Success. *Irish Journal of Academic Practice*, 2(1), 5.
- Eesti elukestva õppe strateegia 2020 (2014). <https://www.hm.ee/sites/default/files/strateegia2020.pdf> (19.03.2018)
- Eesti Euroopa Liidu Poliitika 2015–2019. (2015). <https://riigikantselei.ee/sites/default/files/content-editors/Failid/EL/elpol-2015-2019.pdf> (19.03.2018)
- Eesti infoühiskonna arengukava 2020. (s.a.) https://www.mkm.ee/sites/default/files/elfinder/article_files/eesti_infouhiskonna_arengukava.pdf (07.03.2018)
- Ferrar, A. (2013). DIGCOMP: Kuidas arendada ja mõista digipädevust Euroopas? Digipädevuse enesehindamise raamistik. Luxembourg: Euroopa Liidu Väljaannete Talitus. https://www.hm.ee/sites/default/files/digipadevuse_enesehindamise_raamistik_0.pdf (26.04.2018)

Graham, C. R., Woodfield, W., Harrison, J. B. (2013). A framework for institutional adoption and implementation of blended learning in higher education. *Internet and Higher education*, 18, 4–14.

Griffiths, R., Chingos, M., Mulhern, C., Spies, R. (2014). *Interactive online learning on campus: Testing MOOCs and other platforms in hybrid formats in the University System of Maryland*. http://www.sr.ithaka.org/wp-content/mig/reports/S-R_Interactive_Online_Learning_Campus_20140716.pdf (10.03.2018)

Güzer, B., Caner, H. (2013). The past, present and future of blended learning: an in depth analysis of literature. 5th World Conference on Educational Sciences – WCES 2013. *Procedia Social and Behavioural Sciences* 116, p. 4596–4603.

Gümnaasiumi riiklik õppekava (2011). *Riigi Teataja I*.
<https://www.riigiteataja.ee/akt/129082014021?leiaKehtiv> (07.03.2018)

Hadjerrouit, S. (2008). Towards a blended learning model for teaching and learning computer programming: A case study. *Informatics in Education*, 7(2), p. 181–210.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.7952&rep=rep1&type=pdf> (09.05.2018)

Heidmets, M. (2017). *Õpikäsitus: teooriad, uurimused, mõõtmise. Analüütiline ülevaade*. Tallinna Ülikool, haridusteaduste instituut, lk 11–34.

Holotescu, C., Grosseck, G., Cretu, V., Naaji, A. (2014). Integrating MOOCs in blended courses. In *Proceedings of the 10th International Scientific Conference eLearning and Software for Education*. p. 243–250.

Israel, M. J. (2015). Effectiveness of Integrating MOOCs in Traditional Classrooms for Undergraduate Students. *International Review of Research in Open and Distributed Learning*. Volume 16, Number 5, p. 102–118.
<http://www.irrodl.org/index.php/irrodl/article/view/2222/3402> (14.03.2018)

Jansen, D., Schuwer, R. (2015). Institutional MOOC strategies in Europe. *Status Report Based on a Mapping Survey Conducted in October-December 2014*.
<https://www.surfspace.nl/media/bijlagen/artikel-1763-22974efd1d43f52aa98e0ba04f14c9f3.pdf> (07.03.2018)

Kaur, M. (2012). Blended learning – its challenges and future. 3rd World Conference on Learning, Teaching and Educational Leadership (WCLTA-2012). *Procedia Social and Behavioural Sciences* 93, p. 612–617.

De Kock, A., Slegers, P., Voeten, M.J.M. (2004). New Learning and the Classification of Learning Environments in Secondary Education. *Review of Educational Research Summer 2004*, Vol. 74, No. 2, p. 141–170.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1022.7587&rep=rep1&type=pdf>
(07.03.2018)

Koller, D. (2012). How online courses can form a basis for on-campus teaching. *Forbes*. 2012. <https://www.forbes.com/sites/braintree/2017/11/06/from-brick-and-mortar-to-contextual-commerce-infographic/#1306a3a68b6c> (10.03.2018)

Kori, K., Altin, H., Pedaste, M., Palts, T., Tõnisson, E. (2015a). “What influences students to study information and communication technology?” In: INTED2015 Proceedings: 9th International Technology, Education and Development Conference Madrid, p. 1477–1486.

Kori, K., Pedaste, M., Tõnisson, E., Palts, T., Altin, H., Rantsus, R., Sell, R., Murtazin, K., Rüütmann, T. (2015b). First-year dropout in ICT studies. *Proceedings of the IEEE Global Engineering Education Conference*, Tallinn, Estonia, p. 437–445.
https://sisu.ut.ee/sites/default/files/ikt/files/contribution1140_b.pdf (08.03.2018)

Lage, M., Platt, G., Treglia, M. (2000). “Inverting the classroom: A gateway to creating an inclusive learning environment”. *Journal of Economic Education*, vol. 31(1), p. 30–43.

Lamesoo, K., Ader, A. (2016). Muutunud õpikäsituse rakendamise ja selleks erinevate valikute pakkumise seire korraldamine, sh gümnaasiumis valikkursuste rakendamise osas, projekti raport.
<https://dspace.ut.ee/bitstream/handle/10062/55926/Valikained%20gymn.pdf?sequence=1&isAllowed=y> (07.03.2018)

Leping, V., Lepp, M., Niitsoo, M., Tõnisson, E., Vene, V., Villems, A. (2009). Python Prevails. *Proceedings of the International Conference on Computer Systems and Technologies: CompSysTech'09*, p. IV.17-1-IV.17-5.

Lepp, M., Tõnisson, E. (2015). Integrating Flipped Classroom Approach and Work in Pairs into Workshops in Programming Course. *The 11th International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'15)*, p. 220.

Leppik, C., Haaristo, H-S., Mägi, E. (2017). IKT-haridus: digioskuste õpetamine, hoiakud ja võimalused üldhariduskoolis ja lasteaias. Tallinn: Poliitikauuringute Keskus Praxis. http://www.praxis.ee/wp-content/uploads/2016/08/IKT-hariduse-uuring_aruanne_mai2017.pdf (24.03.2018)

Littlejohn, A., Hood, N., Milligan, C., Mustain, P. (2016). Learning in MOOCs: Motivations and self-regulated learning in MOOCs. *The Internet and Higher Education*, 2016, 29, 40e48.

Mannila, L., de Raadt, M. (2006). An Objective Comparison of Languages for Teaching Introductory Programming. *Proceedings, Koli Calling*. https://eprints.usq.edu.au/1701/3/Mannila_DeRaadt_KOLI2006_PV.pdf (07.03.2018)

Mets, U., Nevsk, E., Pedaste, M., Laanpere, M. (2016). Digipädevus õppekavades. Juhendmaterjal. Hariduse Infotehnoloogia Sihtasutus. https://www.hm.ee/sites/default/files/digipadevusoppekavades_2016veebi.pdf (26.04.2018)

Mohorovičić, S., Strčić, V. (2011). An overview of computer programming teaching methods. *Proceedings of the 22nd Central European Conference on Information and Intelligent Systems*, Faculty of Organization and Informatics Varaždin, Croatia, 2011, p. 47–52.

O'Kelly, J., Bergin, S., Dunne, S., Gaughran, P., Ghent, J., Mooney, A. (2004). An Overview of the Integration of Problem Based Learning into an existing Computer Science Programming Module. *Present at Pleasure By Learning conference*. <https://pdfs.semanticscholar.org/f60e/8ea65110900e1f4c3816bc720591fcc41f79.pdf> (19.03.2018)

Pedaste, M., Kori, K., Tõnisson, E., Palts, T., Altin, H., Rantsus, R. (2016). What Happens to IT Education? The Case in Estonia with Some Recommendations for International Discussion. 2015. https://sisu.ut.ee/sites/default/files/ict/files/pedaste_et_al_2016.pdf (07.03.2018)

ProgeTiiger veebileht. (s.a.). <http://www.progetiiger.ee/tool/53/python> (19.03.2018)

Programmeerimine maalähedaselt. (s.a.). Tartu Ülikool, arvutiteaduse instituut. Kursuse veebileht. <https://courses.cs.ut.ee/2017/progmaa/fall/> (18.03.2018)

Programmeerimise alused. (s.a.). Tartu Ülikool, arvutiteaduse instituut. Kursuse veebileht. <https://courses.cs.ut.ee/2017/eprogalused/spring> (18.03.2018)

Programmeerimise alused II. (s.a.). Tartu Ülikool, arvutiteaduse instituut. Kursuse veebileht (kuupäev puudub) <https://courses.cs.ut.ee/2017/eprogalused2/spring> (18.03.2018)

Programmeerimise alused õpilastele. (s.a.). Tartu Ülikool, arvutiteaduse instituut. Kursuse veebileht. <https://courses.cs.ut.ee/2017/eprogalkool/fall/> (18.03.2018)

Puniste, S. (2014). Eesti gümnaasiumides õpetatavad programmeerimiskursused. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö.
<http://dspace.ut.ee/bitstream/handle/10062/56092/thesis.pdf?sequence=1&isAllowed=y>
(07.03.2018)

Põhikooli Riiklik õppekava. (2011). Riigi Teataja.
<https://www.riigiteataja.ee/aktiis/0000/1327/3133/13275469.pdf> (07.03.2018)

Python. (2018). Python Software Foundation [US]. Ametlik veebileht.
<https://www.python.org/> (26.04.2018)

Python. Jaagup Kippari õppematerjalid. (s.a.).
<http://minitorn.tlu.ee/~jaagup/kool/java/naited20/html/python/juht.html> (14.03.2018)

Python. Kuutõrvaja õppematerjalid. (s.a.). <http://kuutorvaja.eenet.ee/wiki/Python>
(14.03.2018)

Python koolis. (s.a.). Tartu Ülikool, arvutiteaduse instituut. Õppematerjalid.
<https://courses.cs.ut.ee/t/pythonkoolis> (14.03.2018)

Python 3 esimesed sammud. (s.a.). Metshein materjalid.
<https://www.metshein.com/course/python-3-esimesed-sammud/> (14.03.2018)

Sarap, I. (2017). Vaba juurdepääsuga e-kursused kõrgkoolis. Kursuse „Programmeerimise alused” näide. TÜ arvutiteaduse instituudi bakalaureusetöö.

Shah, D. (2015). MOOC Trends in 2015: The Death of Free Certificates. How Much Do Different MOOC Providers Charge for a Certificate in 2015? *Class Central*. December. <https://www.class-central.com/report/death-of-free-certificates/> (02.04.2018)

Smit, K., de Brabander, C.J., Martens, R.L. (2014). Student-centred and teacher-centred learning environment in pre-vocational secondary education: Psychological needs, and motivation, *Scandinavian Journal of Educational Research*, 58:6, 695–712.

Stephenson, C. (2000). A Report On High School Computer Science Education In Five U.S. States. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.195.2415&rep=rep1&type=pdf> (07.03.2018)

Zhang, X., Zhang, C., Stafford, T.F., Zhang, P. (2013). Teaching Introductory Programming to IS Students: The Impact of Teaching Approaches on Learning Performance. *Journal of Information Systems Education*, vol. 24(2), p. 147–155.

Tarrao, S., Tiirmaa-Oras, S. (2007). Blended learning: Research reports & examples of best practices. *B-learn Project. Coordinated by University of Tartu*.

Tartu Ülikooli IT-kursuste materjalide veebileht. (s.a.). <http://programmeerimine.ut.ee/materjalid/> (15.03.2018)

Tartu Ülikooli VÕTA veebileht. (2018) <https://www.ut.ee/et/oppimine/uliopilasele/lisaoigused/vota> (08.03.2018)

TIOBE Index. (2018). TIOBE Software BV veebileht. <https://www.tiobe.com/tiobe-index/> (26.04.2018)

Tutvumine Pythoniga. (s.a.). Kursuse „Rakenduste loomine ja programmeerimise alused” õppematerjal. TTÜ informaatikainstituut. http://scratch.ttu.ee/failid/Python_sisse.pdf (14.03.2018)

Vaikjärv, T., Pilli, E. (2015). Ümberpööratud klassiruumi meetod kui õppija vastutuse kujundaja. *ENDC Proceedings*, Vol. 20, p. 165–175. http://www.ksk.edu.ee/wp-content/uploads/2016/01/KVYOA_20_10_Pilli_Vaikjarv.pdf (26.04.2018)

Õpetajate koolituse raamnõuded. (2015). Riigi Teataja I.

<https://www.riigiteataja.ee/akt/122032011015?leiaKehtiv> (07.03.2018)

Lisad

I. Kursuse „Programmeerimise alused” läbiviimise plaan ja ajakava

Tabel 11. Kursuse teemade läbimise ajakava plaan ja tegelikkus.

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
15.01			Kursuse eeltutvustus. Isikuandmete kasutamise luba! Avaküsitlus	Kursuse eeltutvustus. Lastevanemate load isikuandmete kasutamiseks. Avaküsitlus.
22.01 1. nädal	2	TEEMA: Sissejuhatus 1.1 Algoritm 1.2 Programm 1.3 Andmetüübid 1.4 Muutujad 1.5 Sisend kasutajalt: <i>input()</i> 1.6 Silmaring: Ajaloost 1.7 Lugu: Alkeemia I 1.8 Kontrollülesanded	TEEMA: Sissejuhatus 1.1 Algoritm 1.2 Programm 1.3 Andmetüübid 1.4 Muutujad 1.5 Sisend kasutajalt: <i>input()</i> LISAMATERJAL 1.6 Silmaring: Ajaloost 1.8 Kontrollülesanded (osa ülesandeid BOONUSena)	TEEMA: Sissejuhatus Tunnis: 1.1 Algoritm 1.2 Programm 1.3 Andmetüübid Paaristöö tunni teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 1.1 (Tervitus) Kontrollülesanne 1.2 (President) Kontrollülesanne 1.3 (Astendamine)
29.01 2. nädal	1+2	TEEMA: Tingimuslause 2.1 Tingimuslause 2.2 Tingimuslause tingimuslause sees 2.3 Mitmeosaline tingimus. Loogilised tehted ja avaldised 2.4 Mitmeharuline tingimuslause <i>elif</i> abil 2.5 Juhuslik arv 2.6 Kilpkonnagraafika 2.7 Silmaring: Arvustüsteemid	TEEMA: Tingimuslause 2.1 Tingimuslause 2.2 Tingimuslause tingimuslause sees 2.3 Mitmeosaline tingimus. Loogilised tehted ja avaldised 2.4 Mitmeharuline tingimuslause <i>elif</i> abil 2.10 Kontrollülesanded (2.1, 2.2, 2.3) LISAMATERJAL	TEEMA: Sissejuhatus <i>Eelnevalt kodus: loetud uued teemad!</i> 1.4 Muutujad 1.5 Sisend kasutajalt: <i>input()</i> Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 1.4a (Nädala ajakulu) Kontrollülesanne 1.4b (Kiiruseületamise

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
		2.8 Silmaring: Tõeväärtused 2.9 Lugu: Alkeemia II 2.10 Kontrollülesanded (2.1, 2.2, 2.3) 2.11 Kontrollülesanded (2.4abc)	2.7 Silmaring: Arvusüsteemid	trahv) • TEST-1
5.02 3. nädal	1+2	TEEMA: Tsükel 3.1 Tsükel 3.2 Tsükel. Igal sammul juhtub midagi 3.3 Tsüklist. Mitu korda? 3.4 Juhuslikust arvust veel 3.5 Silmaring: Labürint 3.6 Lugu: Alkeemia III 3.7 Kontrollülesanded 3.1, 3.2, 3.3 3.8 Kontrollülesanded 3.4abc	TEEMA: Tingimuslause 2.5 Juhuslik arv 2.6 Kilpkonnagraafika LISAMATERJAL 2.8 Silmaring: Tõeväärtused 2.11 Kontrollülesanded 2.4abc	TEEMA: Tingimuslause <i>Eelnevalt kodus: loetud uued teemad!</i> 2.1 Tingimuslause 2.2 Tingimuslause tingimuslause sees 2.3 Mitmeosaline tingimus. Loogilised tehted ja avaldised Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: • Kontrollülesanne 2.1 (Jäätumine) • Kontrollülesanne 2.2 (Spämm) • Kontrollülesanne 2.3 (Leedu perenimed)
12.02 4. nädal	1+2	TEEMA: Sõned. Graafika 4.1 Sõned 4.2 Graafikaraamistik tkinter. Tahvel 4.3 Kasutajaliides tkinteriga 4.4 Silmaring: Arvutigraafika 4.5 Kontrollülesanded IV	TEEMA: Tsükel 3.1 Tsükel 3.2 Tsükel. Igal sammul juhtub midagi 3.3 Tsüklist. Mitu korda? 3.4 Juhuslikust arvust veel LISAMATERJAL 3.5 Silmaring: Labürint 3.7 Kontrollülesanded 3.1, 3.2, 3.3 BOONUS: 3.8 Kontrollülesanded 3.4abc	TEEMA: Tingimuslause <i>Eelnevalt kodus: loetud uued teemad!</i> 2.4 Mitmeosaline tingimuslause elif abil 2.5 Juhuslik arv 2.6 Kilpkonnagraafika Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: • Kontrollülesanne 2.4a (Pulss) VÕI Kontrollülesanne 2.4b (Istekoht) VÕI Kontrollülesanne 2.4c (Bussid) • TEST-2
19.02 5. nädal	1+2	TEEMA: Järjend 5.1 Järjend 5.2 For-tsükel, funktsioon <i>range</i> 5.3 Andmed failist 5.4 Silmaring: Krüpteerimine	TEEMA: Sõned 4.1 Sõned 4.5 Kontrollülesanded IV - neist ülesanne 4.1	TEEMA: Tsükel <i>Eelnevalt kodus: loetud uued teemad!</i> 3.1 Tsükel 3.2 Tsükel. Igal sammul juhtub midagi 3.3 Tsüklist. Mitu korda?

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
		5.5 Lugu: Alkeemia IV 5.6 Kontrollülesanded 5.1, 5.2, 5.3 5.7 Kontrollülesanded 5.4abc		3.4 Juhuslikust arvust veel Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 3.1 (Äratus) Kontrollülesanne 3.2 (Laikimine) Kontrollülesanne 3.3 (Täringumäng) Kontrollülesanded 3.4abc - edasijõudnutele TEST-3
		VAHEAEG		
5.03 6. nädal	1+2	TEEMA: Funktsioon 6.1 Funktsioon 6.2 Funktsioonil on väärtus 6.3 Argumendid, muutujad 6.4 Graafika ja funktsioon 6.5 Silmaring: Rekursioon 6.6 Lugu: Alkeemia V 6.7 Kontrollülesanded 6.1, 6.2, 6.3 6.8 Kontrollülesanded 6.4abc	TEEMA: Graafika 4.2 Graafikaraamistik tkinter. Tahvel 4.3 Kasutajaliides tkinteriga 4.5 Kontrollülesanded IV - neist ülesanne 4.2 Lipp LISAMATERJAL 4.4 Silmaring: Arvutigraafika	TEEMA: Sõned. Graafika <i>Eelnevalt kodus: loetud uued teemad!</i> 4.1 Sõned 4.2 Graafikaraamistik tkinter. Tahvel 4.3 Kasutajaliides tkinteriga Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 4.1 (Suured tähed) Kontrollülesanne 4.2a (Lipp) VÕI Kontrollülesanne 4.2b (Liiklusmärk) VÕI Kontrollülesanne 4.2c (Maja) TEST-4
12.03 7. nädal	1+2	TEEMA: Andmevahetus. Lihtne kasutajaliides 7.1 Lugemine veebist 7.2 Lugemine failist 7.3 Faili kirjutamine 7.4 Lihtne kasutajaliides EasyGuiga 7.5 Silmaring: Tekstikodeering 7.6 Lugu: Alkeemia VI	TEEMA: Järjend 5.1 Järjend 5.2 For-tsükkel, funktsioon range 5.3 Andmed failist LISAMATERJAL 5.4 Silmaring: Krüpteerimine	TEEMA: Järjend <i>Eelnevalt kodus: loetud uued teemad!</i> 5.1 Järjend 5.2 For-tsükkel, funktsioon range 5.4 Silmaring: Krüpteerimine Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded:

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
		7.7 Kontrollülesanded 7.1, 7.2, 7.3 7.8 Kontrollülesanded 7.4abc	5.6 Kontrollülesanded 5.1, 5.2, 5.3 5.7 Kontrollülesanded 5.4abc	<ul style="list-style-type: none"> Kontrollülesanne 5.2 (Laikimine <i>for</i>-tsükliga) TEST-5
19.03 8. nädal	1+2	TEEMA: Lõpuks 8.1 Kordamiseks 8.2 Arvestusülesanne 8.3 Loovtöö 8.4 Silmaring: Ametid IT sektoris 8.5 Silmaring: Erinevad programmeerimiskeeled 8.6 Silmaring: Õppimisvõimalused 8.7 Silmaring: Informaatikaolümpiaad 8.8 Silmaring: Regulaaravaldis 8.9 Lugu: Alkeemia VII	TEEMA: Funktsioon 6.1 Funktsioon 6.2 Funktsioonil on väärtus 6.3 Argumendid, muutujad 6.4 Graafika ja funktsioon LISAMATERJAL 6.5 Silmaring: Rekursioon 6.7 Kontrollülesanded 6.1, 6.2, 6.3 6.8 Kontrollülesanded 6.4abc – (<i>kui läheb paljaks, siis hiljem</i>)	TEEMA: Järjend <i>Eelnevalt kodus: loetud uued teemad!</i> 5.3 Andmed failist Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 5.1 (Mootorrattad) Kontrollülesanne 5.3 (Sissetulekud) Kontrollülesanne 5.4a (<i>Jukebox</i>) VÕI Kontrollülesanne 5.4b (Loomulik iive) VÕI Kontrollülesanne 5.4c (Tahvli juurde)
26.03 9. nädal	1+2	– (MOOC on lõppenud)	TEEMA: Andmevahetus 7.1 Lugemine veebist 7.2 Lugemine failist 7.3 Faili kirjutamine 7.7 Kontrollülesanded 7.1, 7.2	TEEMA: Funktsioon <i>Eelnevalt kodus: loetud uued teemad!</i> 6.1 Funktsioon 6.2 Funktsioonil on väärtus 6.3 Argumendid, muutujad 6.4 Graafika ja funktsioon 6.5 Silmaring: Rekursioon Tunnis: Paaristöö loetud teemade kinnistamiseks. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanded 6.1 (Bänner) Kontrollülesanne 6.2 (Teleri suurus) Kontrollülesanne 6.3 (Peo eelarve) TEST-6
2.04 10. nädal	1+2	– (MOOC on lõppenud)	TEEMA: Lihtne kasutajaliides 7.4 Lihtne kasutajaliides EasyGuiga 7.7 Kontrollülesanded 7.3	TEEMA: Andmevahetus <i>Eelnevalt kodus: loetud uued teemad!</i> 7.1 Lugemine veebist

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
				7.2 Lugemine failist Vabatahtlikult edasijõudnutele: 7.3 Faili kirjutamine 7.4 Lihtne kasutajaliides EasyGuiga 7.5 Silmaring: Tekstikodeering Tunnis: Paaristöö loetud teemade kinnistamiseks. Jätkus 8. nädala kohustuslike ülesannete lahendamine. <i>Andmevahetuse teema kohta ülesandeid vabatahtlikud.</i> Vabatahtlikud ülesanded edasijõudnutele: <ul style="list-style-type: none"> Kontrollülesanded 7.1 (Telegramm) Kontrollülesanne 7.2 (Päevik) Kontrollülesanne 7.3 (Kalkulaator) TEST-7
9.04 11. nädal	1+2	– (MOOC on lõppenud)	TEEMA: Lõpuks 7.8 Kontrollülesanded 7.4abc 8.1 Kordamiseks 8.2 Arvestusülesanne	TEEMA: Lõpuks <i>Eelnevalt kodus: loetud uued teemad!</i> 8.1 Kordamiseks Tunnis: Paaristöö loetud teemade kinnistamiseks. Vabatahtlikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanded 6.4a VÕI Kontrollülesanded 6.4b VÕI Kontrollülesanded 6.4c Kontrollülesanne 7.4a VÕI Kontrollülesanne 7.4b VÕI Kontrollülesanne 7.4c
16.04 12. nädal	1+2	– (MOOC on lõppenud)	TEEMA: Lõpuks 8.4 Silmaring: Ametid IT sektoris 8.5 Silmaring: Erinevad programmeerimiskeeled 8.6 Silmaring: Õppimisvõimalused 8.7 Silmaring: Informaatikaolümpiaad	TEEMA: Lõpuks <i>Eelnevalt kodus: loetud uued teemad!</i> 8.4 Silmaring: Ametid IT sektoris 8.5 Silmaring: Erinevad programmeerimiskeeled 8.6 Silmaring: Õppimisvõimalused

Aeg	Tunde	Teema MOOC-is	Plaan	Tegelikkus
			8.8 Silmaring: Regulaaravaldis 8.3 Loovtöö või Arvestustöö Lõpuküsitlus	8.7 Silmaring: Informaatikaolümpiaad 8.8 Silmaring: Regulaaravaldis Kohustuslikud ülesanded: <ul style="list-style-type: none"> 8.2 Arvestusülesanne VÕI 8.3 Loovülesanne
KOKKU	35 t			

II. Kursuse „Programmeerimise alused” ainekava

Õppe- ja kasvatuseesmärgid

Valikkursusega taotletakse, et õppija:

1. arendab loovust, loogilist, analüütilist ja algoritmilist mõtlemist ning süsteemset käsitusviisi probleeme ja ülesandeid lahendades;
2. tunneb programmide loomise vahendeid;
3. omandab programmide koostamise ning probleemide lahendamise baasoskused;
4. saab aru andmete olemusest, nende omadustest ning nendega täidetavatest tegevustest programmides;
5. omandab programmeerimise põhikontseptsioonid ja mõisted ning oskused kavandada, koostada, siluda ja testida programme, mis koosnevad mitmest koostoimivast üksusest (funktsioonist); kasutab arvandmeid ja massiive ning kirjeldab eri liiki protsesse.

Kursuse lühikirjeldus

Kursuse põhiosad:

1. Algoritm. Programm. Andmetüübid.
2. Andmetüübid. Muutujad. Sisend kasutajalt: *input()*.
3. Tingimuslause. Tingimuslause tingimuslause sees. Mitmeosaline tingimus. Loogilised tehted ja avaldised. Mitmeharuline tingimuslause *elif* abil.
4. Juhuslik arv. Kilpkonnagraafika.
5. Tsükkel. Tsükkel – igal sammul juhtub midagi. Tsükkel - mitu korda? Juhuslikust arvust veel.
6. Sõned. Graafikaraamistik tkinter. Tahvel. Kasutajaliides tkinteriga.
7. Järjend. *For*-tsükkel, funktsioon *range*.
8. Andmed failist.
9. Funktsioon. Funktsioonil on väärtus.
10. Argumendid, muutujad. Graafika ja funktsioon.

Õpitulemused

Kursuse lõpul õppija (Programmeerimise alused õpilastele, s.a.):

1. õppija oskab programmeerimise baaskonstruktsioone (tingimuslause, tsükkel, funktsioon) esitada nii plokk skeemidena kui ka programmilõikudena programmeerimiskeeles Python;
2. õppija oskab lihtsa ülesande realiseerida programmina programmeerimiskeeles Python, kasutades seejuures erinevaid baaskonstruktsioone ja andmetüüpe (täisarvud, ujukomaarvud, sõned, järjendid), muutujaid, operatsioone, graafikavahendeid ning andmevahetust failidega;
3. õppija oskab leida ja sõnastada enda eri- või huvialaga seotud programmeerimisülesandeid ja neid lahendada;

4. õppija omab esmast ülevaadet programmeerimise ajaloost, erinevatest programmeerimiskeeltest, arvusüsteemidest, regulaaravaldisest;
5. õppija on mõelnud läbi, kas ta tahab end edaspidi programmeerimisega rohkem kurssi viia ning on teadlik sellekohastest võimalustest.

Õppesisu

Programmeerimise alused. Programmi olemus. Programmide loomise, töötlemise ja täitmise põhimõtted. Programmeerimiskeel Python. Algoritmi olemus. Algoritmide esitusviisid: plokk skeemid. Andmete käsitlemine programmides. Andmete liigid. Operatsioonid andmetega. Muutujad. Avaldised ja funktsioonid. Andmete lugemine failist. Graafikaandmete kasutamine programmides. Graafikaobjektide importimine. Graafiliste kujundite loomine (joonistamine) programmi poolt. Graafikaobjektide põhiomadused ja nende muutmine. Protsesside liigid programmides: järjestikused protsessid, tsüklilised protsessid (kordus), hargnevad protsessid (valik). Eri liiki protsesside kirjeldamise ja täitmise põhimõtted. Korduste liigid: lõpmatu kordus, etteantud kordamisarvuga kordus, eelkontrolliga kordused. Massiivid (järjendid). Massiivide põhiomadused: nimi, dimensioon, mõõtmed, dünaamilisus. Massiivide määramine ja loomine. Viitamine massiivide elementidele. Tüüp algoritmide massiividega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine, sorteerimine jmt. Programmide silumine ja testimine. Programmide vormistamine.

Õppetegevus

1. Õppetegevus toimub kombineeritud õppe vormis: e-õppena ja auditoorse õppena arvutiklassis.
2. Õppetööd toetab elektroonsete õppematerjalide komplekt: teooria põhiosad, harjutused, näited, lingid lisamaterjalidele, õppevideod.
3. Õpilaste iseseisvalt tehtud tööde alusel määratakse hinne.

Füüsiline õpikeskkond

1. Üks osa õppetegevusest toimub õpilase iseseisva tööna. Eelnevalt veenduda, et igal õpilasel on võimalik kasutada arvutit ning internetti. Vastasel juhul võimaldada õpilasel kokkulepitud ajal kasutada arvutitöökohta koolis.
2. Teine osa õppetegevusest toimub arvutiklassis. Üldjuhul on igal õpilasel eraldi arvutitöökoht. Kasutatakse vaba tarkvara ning õpilasele tagatakse turvaline veebipõhine töökeskkond (Moodle, Courses-keskkond), juurdepääs vajalikele infosüsteemidele (e-päevik) ning programmeerimise õpikeskkond (Thonny).

III. Kursuse „Programmeerimise alused” töökava

TÖÖKAVA III/IV KOOLIASTMELE

Õppeaine: Programmeerimise alused

Klass: 9. klass/10. klass

Maht: II poolaasta

Tundide arv: 35 tundi, ühe tunni pikkus 45 min, auditoorsed tunnid paaristundidena

Õppimine toimub MOOC-i kombineeritud õppena.

Õppeprotsessis kasutatakse:

- programmeerimiskeelt **Python**.
- programmeerimiskeskonda **Thonny**, alla laetav aadressil <http://thonny.org>
- e-õppekeskkonda **Moodle** (moodle.ut.ee)

Kasutatav õppekirjandus: MOOC-i õppematerjalid <https://courses.cs.ut.ee/2018/blprogal/spring/Main/HomePage>

Vajalik taristu: arvuti, Internet, e-keskkonnad, e-post

Läbivad teemad: (vt siit <https://www.curriculum.ut.ee/et/opetajalekoolile/labivad-teemad>)

- Elukestev õpe ja karjääri planeerimine – suhtlusoskused e-keskkonnas, enesekontroll, õpi- ja töömeetodid
- Keskkond ja jätkusuutlik areng – tööharjumused ja keskkond
- Kodanikualgatus ja ettevõtlikkus – koostööoskuse arendamine, probleemide lahendamise oskus
- Kultuuriline identiteet – sallivus ja lugupidav suhtumine
- Teabekeskond – internetis käitumine, teabe leidmine internetist, ümbritseva infokeskkonna tajumine ja teadvustamine
- Tehnoloogia ja innovatsioon – kaasaegsete tehnoloogiate eesmärgipärane kasutamine, loovuse arendamine tehnoloogia rakendamisel
- Tervis ja ohutus – ohutuse ja tervise väärtustamine, õiged võtted arvutiga töötamisel
- Väärtused ja kõlblus – lugupidav ja sobilik käitumine internetis, sobivate õpikeskkondade valimine

Õpitulemused kursuse lõpetamisel (Programmeerimise alused õpilastele, s.a.):

- õppija oskab programmeerimise baaskonstruksioone (tingimuslause, tsükkel, funktsioon) esitada nii plokkskeemidena kui ka programmilõikudena programmeerimiskeeles Python;
- õppija oskab lihtsa ülesande realiseerida programmina programmeerimiskeeles Python, kasutades seejuures erinevaid baaskonstruksioone ja andmetüüpe (täisarvud, ujukomaarvud, sõned, järjendid), muutujaid, operatsioone, graafikavahendeid ning andmevahetust failidega;
- õppija oskab leida ja sõnastada enda eri- või huvialaga seotud programmeerimisülesandeid ja neid lahendada;
- õppija omab esmast ülevaadet programmeerimise ajaloost, erinevatest programmeerimiskeeltest, arvusüsteemidest, regulaaravaldisest;
- õppija on mõelnud läbi, kas ta tahab end edaspidi programmeerimisega rohkem kurssi viia ning on teadlik sellekohastest võimalustest.

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
1	Sissejuhatus kursusesse	1	E-keskkonnad Moodle, Courses, Thonny	Kursuse eeltutvustus. Avaküsitlus. E-keskkondade Moodle' ja courses.cs.ut.ee tutvustus ja kasutamiseks ligipääsu loomine, materjalide esitamise struktuuriga tutvumine. Thonny allalaadimise õpetuse tutvustamine, tööpõhimõtted.	1. Oskab kasutada Moodle'i õppekeskkonda, teab, kuidas leida vajalikke õppematerjale ja jälgida oma edenemist. 2. Teab, kuidas pääseda ligi Courses lehe õppematerjalidele ning neid kasutada. 3. Oskab kodus oma arvutisse laadida Thonny.	thonny.org courses.cs.ut.ee moodle.ut.ee
	Algoritm Programm Andmetüübid	1	Algoritm. Programm. Plokkskeem. Andmetüübid (täisarvud, ujukomaarvud, sõne, tõeväärtused). Käsud <i>print</i> ja <i>round</i> .	Õpetaja tutvustab õpitavat teemat, toimub paardesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 1): (andmetüübid <i>string</i> , <i>int</i> , <i>float</i> , <i>bool</i> , käsk <i>print</i> , ümardamine <i>round</i> käsuga). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kontrollülesanded: • Kontrollülesanne 1.1 (Tervitus)	1. Oskab Thonny-ga programmi kirjutada ja käivitada. 2. Oskab Moodle'isse oma programmi üles laadida automaatkontrollile esitamiseks. 3. Mõistab algoritmi olemust ning esitusviisina plokkskeemi koostamist. 4. Tunneb erinevaid	Tööleht 1

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
				<ul style="list-style-type: none"> • Kontrollülesanne 1.2 (President) • Kontrollülesanne 1.3 (Astendamine) 	andmetüüpe.	
2	E-õppe osa	1	Lugeda teemad: <ul style="list-style-type: none"> • Muutujad • Sisend kasutajalt: <i>input()</i> 	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb programmide loomise, töötlemise ja täitmise algseid põhimõtteid.	
	Andmetüübid. Muutujad. Sisend kasutajalt: <i>input()</i> .	2	Tehted arvudega (+liitmine, - lahutamine, / jagamine, * korrutamine, ** astendamine, % jäägi leidmine, // täisosa leidmine) ja sõnedega (liitmine). Muutujale arvulise, tekstilise ja tõeväärtuse andmine ja muutmine. Funktsioonid <i>max</i> ja <i>round</i> .	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 2): (paaristöö, ette antud vigadega programmis vigade leidmine ja parandamine). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kontrollülesanded: <ul style="list-style-type: none"> • Kontrollülesanne 1.4a (Nädala ajakulu) • Kontrollülesanne 1.4b (Kiiruseületamise trahv) • TEST-1 	1. Oskab käsitleda andmeid programmides. 2. Tunneb andmete liike. 3. Oskab teostada operatsioone andmetega. 4. Tunneb muutujaid ning oskab neile väärtusi anda. 5. Oskab kasutajalt andmed küsida ning kasutada muutujatele väärtuste andmisel. 6. Tunneb lihtsamaid matemaatilisi funktsioone.	Tööleht 2
3	E-õppe osa	1	Lugeda teemad: <ul style="list-style-type: none"> • Tingimuslause • Tingimuslause tingimuslause sees • Mitmeosaline tingimus. Loogilised tehted ja avaldised • Mitmeharuline tingimuslause <i>elif</i> abil. 	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb protsesside liike programmides: järjestikused protsessid, tsüklilised protsessid (kordus), hargnevad protsessid (valik). 2. Tunneb eri liiki protsesside kirjeldamise ja täitmise põhimõtteid.	
	Tingimuslause. Tingimuslause tingimuslause sees.	2	Topeltvõrdusmärgi == tähendus. Kooloni olulisus.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 3):	1. Tunneb korduste liike: lõpmatu kordus, etteantud kordamisarvuga kordus,	Tööleht 3

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
	Mitmeosaline tingimus. Loogilised tehted ja avaldised. Mitmeharuline tingimuslause <i>elif</i> abil.		Taande ehk treppimise kasutamine. Hargnemine ehk else-osa. Suurtähtede ja väiketähtede erinevus. Arvude võrdlemine, võrdsuse ja mittevõrdsuse kontrollimine. Programmi kommentaaride tähistamine # märgiga. Loogilised tehted <i>and</i> , <i>or</i> , <i>not</i> abil.	(paaristöö, ette antud töötavate programmide tegevuse mõistmine, mida teeb iga rida). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamisega. Kohustuslikud ülesanded: • Kontrollülesanne 2.1 (Jäätumine) • Kontrollülesanne 2.2 (Spämm) • Kontrollülesanne 2.3 (Leedu perenimed)	eelkontrolliga kordused. 2. Oskab kasutada tsüklilisi protsesse programmides.	
4	E-õppe osa	1	Lugeda teemad: • Juhuslik arv • Kilpkonnagraafika	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb juhusliku arvu mõistet ja kasutamist. 2. Oskab importida kilpkonnagraafika moodulit.	
	Juhuslik arv. Kilpkonnagraafika.	2	Tingimuslause <i>elif</i> abil. Juhusliku arvu genereerimine funktsiooni <i>randint</i> abil. Importimine moodulist <i>random</i> . Moodul <i>turtle</i> ja liikumise käsud <i>forward(n)</i> , <i>back(n)</i> , <i>left(d)</i> , <i>right(d)</i> . Värvimine, käsk <i>fill</i> kasutamine.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 4): (paarisprogrammeerimine, tingimuslause, kilpkonnagraafika). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamisega. Kohustuslikud ülesanded: • Kontrollülesanne 2.4a (Pulss) VÕI Kontrollülesanne 2.4b (Istekoht) VÕI Kontrollülesanne 2.4c (Bussid) • TEST-2	1. Oskab töötada meeskonnas ja võtta erinevaid rolle.	Tööleht 4
5	E-õppe osa	1	Lugeda teemad: • Tsükel	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb protsesside liike programmides: järjestikused	

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
			<ul style="list-style-type: none"> Tsükel. Igal sammul juhtub midagi Tsüklist. Mitu korda? Juhuslikust arvust veel 		protsessid, tsüklilised protsessid (kordus), hargnevad protsessid (valik). 2. Oskab eri liike protsesse kirjeldada ja tunneb nende täitmise põhimõtteid.	
	Tsükel. Tsükel - igal sammul juhtub midagi. Tsükel - mitu korda? Juhuslikust arvust veel.	2	<i>While</i> -tsükel, tsüklimuutuja ehk loendaja kasutamine ja väärtuse muutmine lühemate tehtega +=, -=. Kilpkonnagraafika <i>while</i> -tsükliga. Programmi uinutamine <i>sleep</i> käsuga.	Õpetaja tutvustab õpitavat teemat, toimub paardesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 5): (paaristöö, ette antud programmide arusaamine, mis väljastatakse ekraanile, <i>while</i> -tsükel, kilpkonn). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kohustuslikud ülesanded: <ul style="list-style-type: none"> Kontrollülesanne 3.1 (Äratus) Kontrollülesanne 3.2 (Laikimine) Kontrollülesanne 3.3 (Täringumäng) Kontrollülesanded 3.4a VÕI Kontrollülesanne 3.4b VÕI Kontrollülesanne 3.4c TEST-3 	1. Oskab kasutada graafikaandmeid programmides. 2. Oskab importida graafikaobjekte. 3. Oskab luua (joonistada) graafilisi kujundeid programmiga.	Tööleht 5
6	E-õppe osa	1	Lugeda teemad: <ul style="list-style-type: none"> Sõned Graafikaraamistik tkinter. Tahvel Kasutajaliides Tkinteriga 	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb graafikaobjektide põhiomadusi. 2. Tunneb sõne mõistet.	
	Sõned. Graafikaraamistik tkinter. Tahvel. Kasutajaliides	2	Sõnede korral jutumärkide ja ülakomade kasutamine. Langkriips \ ja kaldkriips / ning nende kasutamine	Õpetaja tutvustab õpitavat teemat, toimub paardesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 6): (paaristöö, ette antud programmide arusaamine,	1. Oskab programmis kasutada sõnesid ning nende meetodeid. 2. Oskab sooritada sõnedega	Tööleht 6

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
	tkinteriga.		sõnade juures. Reavahetuse kombinatsioon \n. Sõnemeetodid <i>capitalize()</i> , <i>lower()</i> , <i>upper()</i> , <i>startswith()</i> , <i>endswith()</i> , <i>title()</i> . Sõne sümbolite eraldamine indeksitega. Sõnade korrutamine. Graafikaraamistik tkinter.	mis väljastatakse ekraanile, sõned, graafika). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kohustuslikud ülesanded: <ul style="list-style-type: none"> • Kontrollülesanne 4.1 (Suured tähed) • Kontrollülesanne 4.2a (Lipp) VÕI Kontrollülesanne 4.2b (Liiklusmärk) VÕI Kontrollülesanne 4.2c (Maja) • TEST-4 	lihtsamaid operatsioone. 3.Tunneb graafikaraamistikku tkinter.	
7	E-õppe osa	1	Lugeda teemad: <ul style="list-style-type: none"> • Järjend • <i>For</i>-tsükkel, funktsioon <i>range</i> • Silmaring: Krüpteerimine 	Materjalides olevate enesekontrollitestide lahendamine.	1. Tunneb massiivi järjend. 2. Tunneb massiivide põhiomadusi järjendite näitel: nimi, dimensioon, mõõtmel, dünaamilisus. 3. Oskab järjendeid määratleda ja luua. 4. Oskab viidata järjendi elementidele	
	Järjend. <i>For</i> -tsükkel, funktsioon <i>range</i> .	2	Andmestruktuur järjend ehk list ja indeksite kasutamine. Viilutamine. Elemendi lisamine järjendisse <i>append</i> funktsiooniga. <i>For</i> -tsükkel ja selle kasutamine. Funktsioon <i>range</i> ja tema kasutusvõimalused.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 7): (paaristöö, ette antud programmide arusaamine ja täiendamine, kui on vaja midagi väljastada, järjendid, <i>for</i> -tsükkel, funktsioon <i>range</i>). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kohustuslikud ülesanded: <ul style="list-style-type: none"> • Kontrollülesanne 5.2 (Laikimine for-tsükliga) • TEST-5 	1. Oskab tüüp algoritmide massiividega: summade ja keskmiste arvutamine, ekstreemumite leidmine, otsimine, sorteerimine jmt.	Tööleht 7

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
8	E-õppe osa	1	Lugeda teemad: • Andmed failist	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb andmete failist lugemise võimalust.	
	Andmed failist.	2	Thonnyga tekstifaili koostamine. <i>For</i> -tsükli kasutamine failist andmete lugemiseks. Kilpkonna translaator.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 8): (paarisprogrammeerimine, tekstifailist andmete lugemine <i>for</i> -tsükliga, andmete lugemine järjendisse). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kohustuslikud ülesanded: • Kontrollülesanne 5.1. (Mootorrattad) • Kontrollülesanne 5.3. (Sissetulekud) • Kontrollülesanne 5.4a (Jukebox) • Kontrollülesanne 5.4b (Loomulik iive) • Kontrollülesanne 5.4c (Tahvli juurde)	1. Oskab töötada meeskonnas ja võtta erinevaid rolle. 2. Oskab kasutada <i>for</i> -tsükli andmete lugemisel failist. 3. Oskab Thonnys koostada tekstifaili ja sealt andmeid lugeda. 4. Oskab kasutada järjendit andmete kogumiseks.	Tööleht 8
9	E-õppe osa	1	Funktsioon. Funktsioonil on väärtus.	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb funktsiooni mõistet ja funktsiooni väärtuse mõistet.	
	Funktsioon. Funktsioonil on väärtus.	2	Funktsioon. Funktsiooni defineerimine.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 9): (paarisprogrammeerimine, andmete lugemine tekstifailist – 8. nädala kordamine). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Kohustuslikud ülesanded: • Kontrollülesanne 6.1. (Banner)	1. Oskab töötada meeskonnas ja võtta erinevaid rolle. 2. Oskab kirjutada programmi, mis loeb andmeid failist. 3. Tunneb funktsiooni mõistet. 4. Oskab defineerida funktsiooni.	Tööleht 9

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
				<ul style="list-style-type: none"> • Kontrollülesanne 6.2. (Teleri suurus (tollides)) • Kontrollülesanne 6.3. (Peo eelarve) 		
10	E-õppe osa	1	Argumendid, muutujad. Graafika ja funktsioon	Materjalides olevate enesekontrollitistide lahendamine.	1. Tunneb argumendi ja muutuja mõistet ning kasutamise võimalusi.	
	Argumendid, muutujad. Graafika ja funktsioon. Andmevahetus (edasijõudnutel)	2	Funktsioon. Funktsiooni defineerimine. Rekursiooni mõiste. Käsuriida. Funktsiooni väärtuse tagastamine ja väljastamine. Mitu argumenti. Edasijõudnutel: Faili kirjutamine. Lihtne kasutajaliides EasyGuiga. Silmaring: Tekstikodeering	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 10): (etteantud programmide arusaamine, mis väljastatakse ekraanile, funktsioonide mõistmine ja kasutamine). Kui paarilised on lõpetanud paaristöö ülesanded ning lahenduste paberid õpetajale andnud, alustavad nad programmeerimisülesannete lahendamise. Edasijõudnutele/kiirematele on lisaülesanded (Tööleht – lisa) andmevahetuse kohta. Kohustuslikud ülesanded: <ul style="list-style-type: none"> • Kontrollülesanne 7.1. Telegramm • Kontrollülesanne 7.2. Päevik • Kontrollülesanne 7.3. Kalkulaator 	1. Oskab koostada funktsiooni, määrata funktsiooni argumente, tagastada funktsiooni väärtust ning saadud väärtust kasutada. 2. Tunneb rekursiooni mõistet.	Tööleht 10 Tööleht - lisa
11	E-õppe osa	1	Kordamiseks	Materjalides olevate enesekontrollitistide lahendamine.	1. Oskab oma õppimist juhtida.	
	Lõpuks	2	Järjend. Tingimuslause. Kilpkonnagraafika. Failist andmete lugemine. Funktsioonid.	Õpetaja tutvustab õpitavat teemat, toimub paaridesse jaotumine. Iga paar saab töölehe. Paaristöö ülesanded (Tööleht 11): (Programmis on read sassi aetud, tõsta read sobivatesse kohtadesse nii, et programm töötaks. Programmi tegevusest arusaamine.). Paaristöö kestus kaks tundi. Tööleht on tarvis välja printida, vajalikud abivahendid on käärid, liim.	1. Oskab lihtsa ülesande realiseerida programmina. 2. Tunneb erinevaid baaskonstruksioone, andmetüüpe, muutujaid, operatsioone, graafikavahendeid ning failist andmete lugemist.	Tööleht 11

Nädal	Õppeteema	Tundide arv*	Põhimõisted ja alateemad	Teoreetiline ja praktiline tegevus tunnis	Soovitavad õpitulemused ja hindamine	Kasutatav lisamaterjal veebis/keskkond
					3. Oskab programme esmasel tasemel siluda ja testida. 4. Oskab programme vormistada.	
12	E-õppe osa	1	Lõpuks (Silmaringi teemad: Ametid IT sektoris. Erinevad programmeerimiskeeled. Õppimisvõimalused. Informaatikaolümpiaad. Regulaaravaldis)	Materjalide lugemine	1. Omab esmast ülevaadet programmeerimise ajaloost, erinevatest programmeerimiskeeltest, arvusüsteemidest, regulaaravaldisest. 2. On mõelnud läbi, kas ta tahab end edaspidi programmeerimisega rohkem kurssi viia ning on teadlik sellekohastest võimalustest.	
	Lõpuks	2	Seni õpitud oskuste kasutamine.	Õpetaja poolt antud arvestustöö ülesande lahendamine. Või Õpilase enda poolt leitud teemal õpetaja poolt etteantud tingimustele vastava ülesande koostamine ja lahendamine.	1. Oskab lihtsa ülesande realiseerida programmina, kasutades seejuures erinevaid baaskonstruksioone ja andmetüüpe (täisarvud, ujukomaarvud, sõned, järjendid), muutujaid, operatsioone, graafikavahendeid ning andmevahetust failidega.	

* 1 tund = 45 minutit

IV. Kursuse avaküsitluse küsimustik

Avaküsitlus

Õppeaine „Programmeerimise alused" kursuse eelküsitlus.

Saadud vastuseid kasutatakse kursuse paremaks muutmiseks anonüümselt.

Küsimused on ühel lehel ja neile vastamine võtab aega u 10 minutit.

Palun vasta allpool toodud küsimustele.

Ees- ja perekonnanimi (tüüp: lühivastus):

1. Klass (tüüp: valikküsimus):

- 9. klass
- 10. klass

2. Kuidas oled varem programmeerimisega kokku puutunud? (tüüp: valikküsimus, mitu valikut)

- Pole kunagi programmeerimisega kokku puutunud.
- Olen iseseisvalt proovinud programmeerimist õppida.
- Olen osalenud programmeerimise kursusel/huviringis.
- Tuttavate/sugulaste/sõprade hulgas on programmeerijaid.
- Muu

2.1 Soovi korral kirjelda oma varasemat kokkupuudet programmeerimisega: (tüüp: lühivastust)

3. Kelle poole on Sul võimalus pöörduda programmeerimise küsimustega? (tüüp: lühivastus)

4. Mis motiveeris Sind programmeerimist õppima? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

1. Saan selle kursuse abil enam teadmisi.
2. Saan tulevikus paremini konkureerida tööturul.
3. Mul on huvi selle teema vastu.
4. Saan valikkursuse aine.
5. Saan ennast arendada.
6. Saan endale esitada väljakutse.
7. Sõber tuli õppima ja tulin temaga kaasa.
8. Minu vanemad arvavad, et peaksin seda õppima.

9. Saan alusteadmised, et programmeerimisega edasi tegeleda.
10. Saan alusteadmised, et seda eriala hiljem õppima asuda.
11. Programmeerimine on populaarne ja tahan sellega kursis olla.
12. Õpetaja soovitas.

5. Mida arvad programmeerimisest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

1. Arvan, et saan programmeerimisega hästi hakkama.
2. Arvan, et programmeerimine võib mulle olla sobiv ala.
3. Arvan, et tegelen programmeerimise õppimisega ka tulevikus.
4. Arvan, et programmeerimise oskus on mulle tulevikus vajalik.
5. Arvan, et programmeerimise oskus tuleb kõigile kasuks.

6. Kas sulle meeldib mängida arvutimänge? (tüüp: skaala, 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

7. Kas sulle meeldib matemaatika? (tüüp: skaala, 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

8. Millised on sinu soovid sellelt kursuselt? (tüüp: lühivastus)

V. Kursuse lõpuküsitluse küsimustik

Lõpuküsitlus

Õppeaine „Programmeerimise alused“ kursuse lõpuküsitlus.

Saadud vastuseid kasutatakse kursuse paremaks muutmiseks anonüümselt.

Küsimused on teemade kaupa neljal lehel ja neile vastamine võtab aega u 20 minutit.

Palun vasta allpool toodud küsimustele.

LEHT 1. Andmed

Ees- ja perekonnanimi (tüüp: lühivastus):

Klass (tüüp: valikküsimus):

- 9. klass
- 10. klass

Sugu (tüüp: valikküsimus):

- Mees
- Naine

LEHT 2. Ajaline ülesehitus

1. Mida arvad kursuse ajalisest ülesehitusest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Mulle sobis, et kursuse kestus oli 12 nädalat.
- Ma arvan, et kursuse kestus võiks olla rohkem kui 12 nädalat (sama materjali mahuga).
- Ma arvan, et kursuse kestus võiks olla vähem kui 12 nädalat (sama materjali mahuga).
- Mulle sobis koolis programmeerimine paaristunnina (2 tundi järjest).
- Mulle sobis, et paaristööd tegime esimese tunni ajal ja iseseisvaid ülesandeid lahendasime teise tunni ajal.
- Üldiselt ma olen rahul kursuse korraldusega.

Kui vastasid, et kursuse pikkus võiks olla 12 nädalast erinev, siis mitu nädalat? (tüüp: lühivastus)

2. Keskmiselt oli ette nähtud ühes nädalas 45 minutit iseseisva tööna kodus. Mul kulus keskmiselt ... (minutit). (tüüp: lühivastus, nõue: Sisesta arv (minutites))

3. Keskmiselt oli ette nähtud ühes nädalas 90 minutit koolitunnina. Mul kulus kohustuslike ülesannete lahendamiseks nädalas keskmiselt ... (minutit). (tüüp: lühivastus, nõue: Sisesta arv)

LEHT 3. Tegevused

1. Mida arvad materjalidest ja kodustest tegevustest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Katsetasin läbi kõik materjalides olnud näiteprogrammid.
- Enesekontrolli küsimused materjalide juures olid mulle kasulikud.
- Vaatasin kõiki õppevideoid.
- Programmeerimise materjalid olid tekstiliselt liiga pikad.
- Silmaringi materjale oli huvitav lugeda.

2. Mida arvad tunnis tehtud tegevustest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Paarilisega koos arutamine aitas mul teemast aru saada.
- Paaristöö ülesanded olid mulle kasulikud programmeerimisülesannete lahendamiseks.
- Mulle sobis paaristöö.
- Kohustuslikud programmeerimisülesanded olid mulle keerulised.
- Kohustuslikke programmeerimisülesandeid oli minu jaoks liiga palju.

3. Palun märkida, mis järjekorras Teie harilikult ühe nädala teema tegevusi alustasite. (tüüp: *grid*, skaala 0 – ei teinud, 1 – esimesena, ..., 8 – viimasena)

- Videote vaatamine
- Tunnis tehtud paarisülesannete lahendamine
- Programmeerimise materjalide lugemine
- Programmeerimise kohustuslike ülesannete lahendamine
- Nädala test
- Murelahendajate kasutamine
- Mujalt lisamaterjali otsimine
- Silmaringi materjalide lugemine

4. Mida arvad e-õppe tegevustest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Mulle sobis, et uue materjali pidi läbi töötama enne tundi.
- Mulle meeldis, et sain enne tundi ise valida aja materjalide läbitöötamiseks (nt. alustada ühel päeval ja lõpetada teisel).
- Ma arvan, et kui tunniks mitte valmistuda, on tunnis raskem teemast aru saada.
- Mulle oleks meeldinud, kui ise materjalide lugemise asemel oleks õpetaja tunnis uut teemat seletanud.
- Mulle oleks meeldinud, kui koduse lugemise asemel oleks olnud lisatund koolis.

LEHT 4. Üldiselt kursusest

1. Mida arvad oma õppimisest kursusel? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Süvenesin programmeerimise teemadesse põhjalikult.
- Tundsin õppimisest naudingut.
- Kursuse läbimine oli mulle sobivaks väljakutseks.
- Teemade valik sobis mulle hästi.
- Küsisin abi, kui jäin ülesannetega hätta.
- Ma õppisin sellel kursusel palju.
- Ma arvan, et selle kursuse läbimiseks on vajalikud eelnevad programmeerimisteadmised.

2. Mida arvad programmeerimisest? (tüüp: *grid*, skaala 1 – ei nõustu üldse, ..., 5 – nõustun täielikult)

- Ma arvan, et sain programmeerimisega hästi hakkama.
- Ma arvan, et programmeerimine on mulle sobiv ala.
- Ma arvan, et jätkan programmeerimise õppimist ka tulevikus.
- Ma arvan, et kui sellele kursusele tuleks samalaadne jätkukursus, siis ma osaleksin.
- Kui peaksin praegu valima edasiõppimise/eluala, siis valiksin suure tõenäosusega programmeerimisega seotud ala.
- Arvan, et programmeerimine sobib eelkõige matemaatilise mõtteviisiga inimestele.
- Ma arvan, et minu jaoks kursus õnnestus.

3. Kas soovid veel midagi lisada? (tüüp: lühivastus)

VI. Programmeerimisülesannete tabel

Tabel 12. Kasutatud programmeerimisülesannete kasutatud lühendite selgitused, lahendanute arvud ning lingid ülesannete tekstidele.

Nädal	Lühend	Ülesanne	Arv*	Link ülesande tekstile
1	Ü1 1.1	Kontrollülesanne 1.1. Tervitus	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-sissejuhatus/#yl1.1
	Ü1 1.2	Kontrollülesanne 1.2. President	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-sissejuhatus/#yl1.2
	Ü1 1.3	Kontrollülesanne 1.3. Astendamine	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-sissejuhatus/#yl1.3
2	Ü1 1.4a	Kontrollülesanne 1.4a. Nädala ajakulu	14	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-sissejuhatus/#yl1.4a
	Ü1 1.4b	Kontrollülesanne 1.4b. Kiiruseületamise trahv	13	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-sissejuhatus/#yl1.4b
3	Ü1 2.1	Kontrollülesanne 2.1. Jäätumine	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause/#yl2.1
	Ü1 2.2	Kontrollülesanne 2.2. Spämm	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause/#yl2.2
	Ü1 2.3	Kontrollülesanne 2.3. Leedu perenimed	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause/#yl2.3
4	Ü1 2.4a	Kontrollülesanne 2.4a. Pulss	9	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause2#yl2.4a
	Ü1 2.4b	Kontrollülesanne 2.4b. Istekoht	3	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause2#yl2.4b
	Ü1 2.4c	Kontrollülesanne 2.4c Busside logistika	9	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-tingimuslause2#yl2.4c

Nädal	Lühend	Ülesanne	Arv*	Link ülesande tekstile
5	Ü1 3.1	Kontrollülesanne 3.1. Äratus	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel#yl3.1
	Ü1 3.2	Kontrollülesanne 3.2. Laikimine	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel#yl3.2
	Ü1 3.3	Kontrollülesanne 3.3. Täringumäng	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel#yl3.3
	Ü1 3.4a	Kontrollülesanne 3.4a Laikimine ver. 2	11	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel2#yl3.4a
	Ü1 3.4b	Kontrollülesanne 3.4b Vabavisked	5	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel2#yl3.4b
	Ü1 3.4c	Kontrollülesanne 3.4c Male	7	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-whiletsykkel2#yl3.4c
6	Ü1 4.1	Kontrollülesanne 4.1. Suured tähed	11	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-graafika#yl4.1
	Ü1 4.2a	Kontrollülesanne 4.2a. Eesti haldusüksuse lipp	12	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-graafika#yl4.2a
	Ü1 4.2b	Kontrollülesanne 4.2b. Liiklusmärk	4	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-graafika#yl4.2b
	Ü1 4.2c	Kontrollülesanne 4.2c. Maja	3	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-graafika#yl4.2c
7	Ü1 5.2	Kontrollülesanne 5.2. Laikimine (for-tsükliga)	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid#yl5.2
8	Ü1 5.1	Kontrollülesanne 5.1. Mootorrattad	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid#yl5.1
	Ü1 5.3	Kontrollülesanne 5.3. Sissetulekud	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid#yl5.3
	Ü1 5.4a	Kontrollülesanne 5.4a Jukebox	13	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid2#yl5.4a

Nädal	Lühend	Ülesanne	Arv*	Link ülesande tekstile
	Ü1 5.4b	Kontrollülesanne 5.4b Loomulik iive	2	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid2#yl5.4b
	Ü1 5.4c	Kontrollülesanne 5.4c Tahvli juurde	5	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-jarjendid2#yl5.4c
9	Ü1 6.1	Kontrollülesanne 6.1. Bänner	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-funktsioon#yl6.1
	Ü1 6.2	Kontrollülesanne 6.2. Teleri suurus (tollides)	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-funktsioon#yl6.2
	Ü1 6.3	Kontrollülesanne 6.3. Peo eelarve	15	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-funktsioon#yl6.3
10**	Ü1 7.1	Kontrollülesanne 7.1. Telegramm	3	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-failistfaili#yl7.1
	Ü1 7.2	Kontrollülesanne 7.2. Päevik	3	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-failistfaili#yl7.2
	Ü1 7.3	Kontrollülesanne 7.3. Kalkulaator	3	https://courses.cs.ut.ee/2018/blprogal/spring/Main/Kontroll-failistfaili#yl7.3

* õppijate arv, kes lahendas vastavat ülesannet

** Nädal 10 ülesannete lahendamine ei olnud kohustuslik

VII. Lihtlitsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Kersti Kivisoo** (töö autor),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

Pythoni programmeerimise algõppe e-kursuse „Programmeerimise alused” läbiviimine kombineeritud õppena koolis,

mille juhendajad on Tauno Palts ja Merilin Säde,

1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
 3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **21.05.2018**